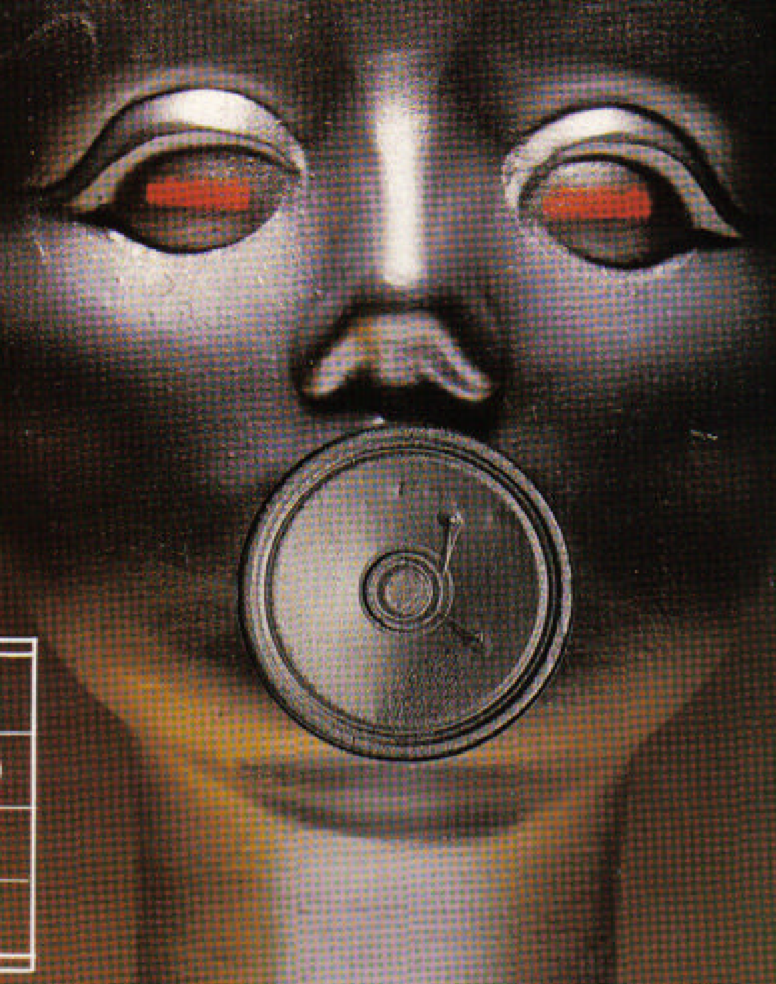


Publicación práctica para usuarios de

commodore

Marzo 1986 Precio 350 Ptas Número 7



MAPA DE MEMORIA C-64

FORMAS DE LA NATURALEZA

CONSTRUYE ESTE SINTETIZADOR DE VOZ

RELACION DE PREMIADOS
con 50.000 ptas (en material)
con 25.000 ptas (en material)
y 10.000 ptas (en material)

CUADRO DE UTILIZACION DEL SPO256

En el ejemplo se ha puesto en mayuscula el alofono correspondiente en una palabra (obligadamente en ingles) propuesta por el fabricante.

DIRECCION	ALOFONO	EJEMPLO EN PALABRA
0	Pausa 10 ms	Utilizar tras detenciones con voz y africadas
1	Pausa 30 ms	" " " " " "
2	Pausa 50 ms	Antes de paradas sin voz y fricativas con voz
3	Pausa 100 ms	" " " " " " " tambien entre palabras
4	Pausa 200 ms	Entre frases
5	OY	bOY
6	AY	skY
7	EH	End
8	KK3	Comb
9	PP	Pit
10	JH	dodGe
11	NN1	thiN
12	IH	sIt
13	TT2	To
14	RR1	Rural
15	AX	sUcceed
16	MM	Milk
17	TT1	parT
18	DH1	THey
19	IY	sEE
20	EY	bEIge
21	DD1	coulD
22	UW1	tO
23	AO	AUght
24	AA	hOt
25	YY2	Yes
26	AE	hAt
27	HH1	He
28	BB1	daB
29	TH	THin
30	UH	bOOk
31	UW2	fOOd
32	AW	OUt
33	DD2	Do
34	GG	wiG
35	W	Vest
36	EF1	GUest
37	SH	SHip
38	ZH	aZure
39	RR2	bRain
40	FF	Food
41	KK2	sKy
42	KK1	Can't
43	ZZ	Zoo
44	NG	aNchore
45	LL	Lake
46	WW	Wool
47	XR	repaiR
48	WH	WHile
49	YY1	Yes
50	CH	CHurch
51	ER1	summER
52	ER2	bURn
53	OW	nOW
54	DH2	THey
55	SS	veST
56	NN2	No
57	HH2	Hoe
58	OR	stORe
59	AR	alArm
60	YR	cleaR
61	EG2	Got
62	EL	saddle
63	BB2	Business

Vocales cortas:7,12,15,23,24,26,30
Vocales largas:5,6,19,20,22,31,32,53,62
Vocales coloreadas:47,51,52,58,59,60
Resonantes:14,39,49,25,45,46
Fricativas con voz:18,54,35,38,43
Fricativas sin voz (hay que utilizarlas dos veces cuando van en inicio):29,40,55
Detenciones con voz:21,28,33,34,36,61,63
Detenciones con voz:8,9,13,17,41,42
Africadas:10,50
Nasales:11,16,44,56

ne para nosotros, esos alófonos almacenados en una ROM interna del *chip* corresponden a la fonética inglesa y se supone que combinándolos adecuadamente se puede producir la síntesis de cualquier palabra. Por lo que a nosotros respecta parece que así es, pero cuando habla castellano con su acento metálico característico parece un turista recién llegado de California. Sin embargo este inconveniente no le resta interés al montaje.

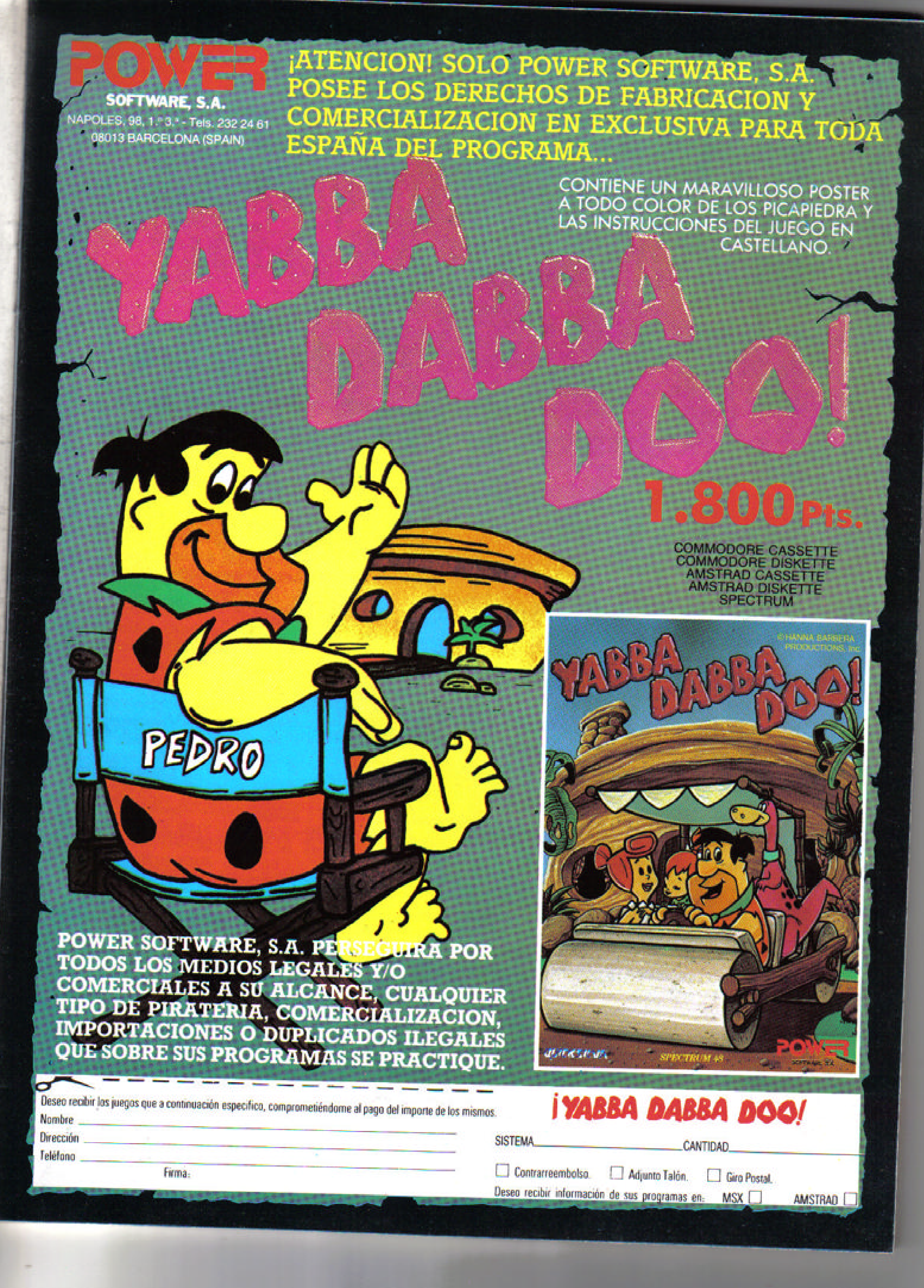
El circuito sintetizador que proponemos es controlado directamente desde el **64** a través del *port* del usuario, tomando de él incluso la corriente de alimentación. El *port* del usuario del ordenador está unido a la segunda CIA, que se encarga de distintas funciones de Entrada/Salida del ordenador. En lo que nos interesa hemos de centrarnos en cuatro registros, a los que se accede como si de posiciones de memoria se tratase. Los cuatro se desglosan en dos régistros de datos y dos registros de dirección de esos datos. Cada uno de los ocho bits de los registros de dirección de datos (DDR) se encargan de definir si el número del bit correspondiente en el registro de datos realiza función de entrada o salida. Esto quiere decir que definen si una línea del *port* recibe señales desde el exterior o, por el contrario, las envía. Cabe aclarar que en el *port* del usuario del **64** en realidad existen dos *ports*, el A y el B. Por tanto los registros de dirección son bautizados como DDRA y DDRB. Si por ejemplo, la información depositada en el DDRA es 255 (todo unos lógicos) el *port* A actúa como elemento de salida de inforamción. Si la información depositada fuera 16, los cuatro bits de menos peso del *port* A asumen funciones de salida, mientras que los cuatro superiores los hacen de entrada.

Aclarado el funcionamiento de los *ports*, sólamente queda decir cuáles son las direcciones de la memoria tras las que se esconden los cuatro registros mencionados: 56576=Registro del *port* A, 56577=Registro del *port* B, 56578=Registro de dirección de datos A y 56579=Registro de dirección de datos B.

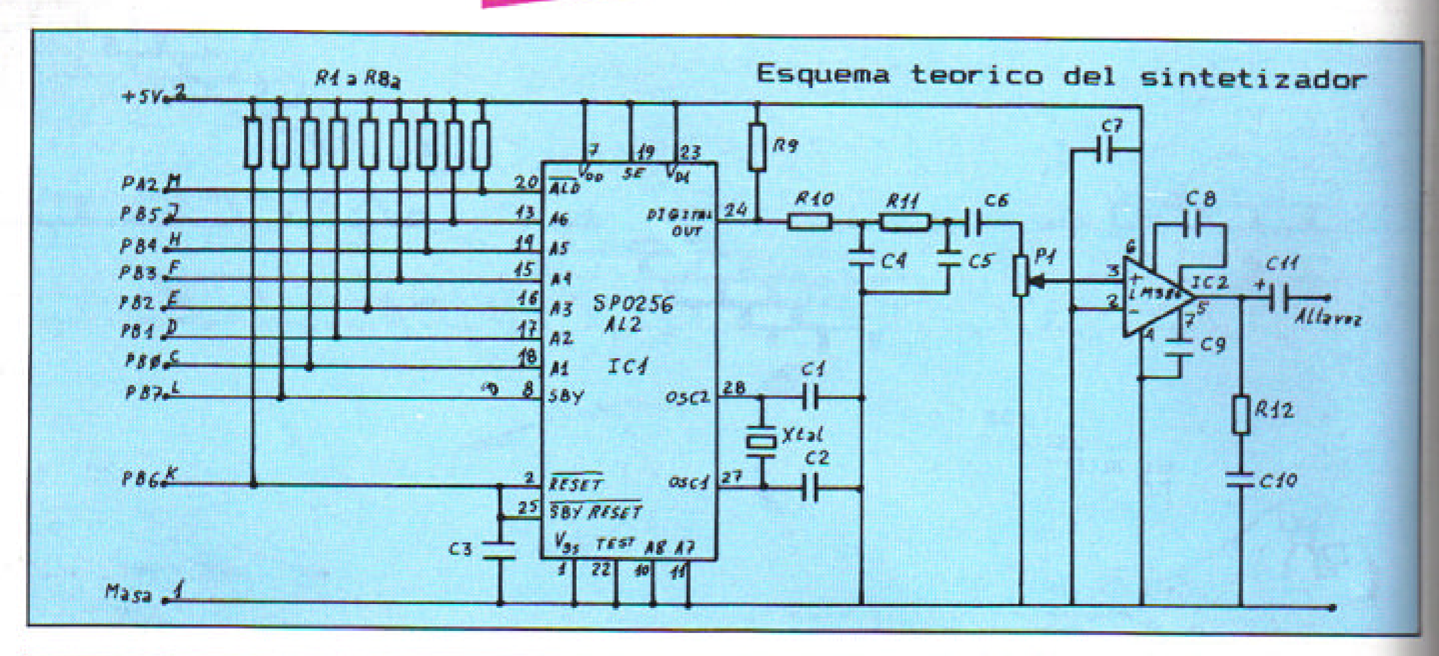
El **SPO256** dispone de seis líneas de direccionamiento, definidas como A1/A6, y a través de ellas se direccionan los sesenta y cuatro alófonos cuya información está contenida en los 2K de memoria ROM existente dentro del *chip*. Este direccionamiento lo realizaremos desde el *port* B, utilizando las líneas PB0 a PB5.

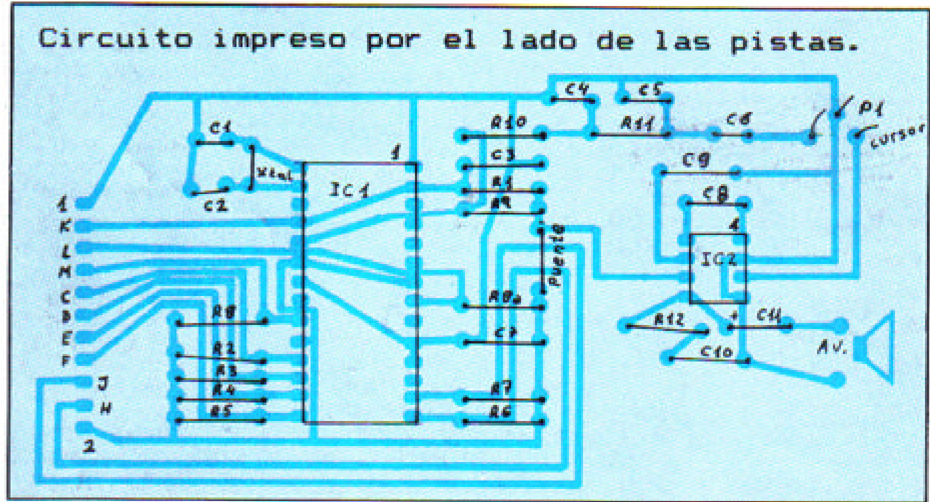
Existe una patilla de puesta a cero (vuelta al estado inicial) en el *chip* sintetizador, ésta lleva la leyenda RESET. PB6 está conectada a ella para su control. Esta función actúa cuando se pone la patilla a cero.

La patilla 8 está conectada a PB7, que es la señal SBY (*Standby*) y es la salida de estado del **SPO256**, toma la misión de informar en que momento



POWER
SOFTWARE, S.A.
NAPOLES, 98, 1.º 3.ª - Tels. 232 24 61
08013 BARCELONA (SPAIN)
¡ATENCION! SOLO POWER SOFTWARE, S.A. POSEE LOS DERECHOS DE FABRICACION Y COMERCIALIZACION EN EXCLUSIVA PARA TODA ESPAÑA DEL PROGRAMA...
CONTIENE UN MARAVILLOSO POSTER A TODO COLOR DE LOS PICAPIEDRA Y LAS INSTRUCCIONES DEL JUEGO EN CASTELLANO.
YABBA DABBA DOO!
1.800 Pts.
COMMODORE CASSETTE
COMMODORE DISKETTE
AMSTRAD CASSETTE
AMSTRAD DISKETTE
SPECTRUM
PEDRO
© HANNA BARBERA PRODUCTIONS, Inc
YABBA DABBA DOO!
SPECTRUM 48
POWER
POWER SOFTWARE, S.A. PERSEGUIRA POR TODOS LOS MEDIOS LEGALES Y/O COMERCIALES A SU ALCANCE, CUALQUIER TIPO DE PIRATERIA, COMERCIALIZACION, IMPORTACIONES O DUPLICADOS ILEGALES QUE SOBRE SUS PROGRAMAS SE PRACTIQUE.
Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.
Nombre
Dirección
Teléfono
Firma:
¡YABBA DABBA DOO!
SISTEMA
CANTIDAD
Contrarreembolso
Adjunto Talón.
Giro Postal.
Deseo recibir información de sus programas en: MSX AMSTRAD





es posible acceder al *chip* porque no está trabajando. Cuando tiene valor 0 el *chip* no puede ser accedido y cuando es 1 el registro de direcciones puede ser cargado con una nueva, es similar a la línea *Acknowledge* en el *interface* **Centronics**. La línea PB7 recoge ésta información.

La patilla 20 recibe la etiqueta ALD (*address load*). Cuando se le aplica un impulso negativo, el registro de direcciones del *chip* memoriza la dirección que en ese preciso instante está presente en las líneas A1/A6. Como se han terminado las líneas del *port* B en el **64**, recurrimos a PA2, línea que dispone de unas características propias que le hacen especialmente adecuada para la función que le asignamos.

Una vez que el registro interno correspondiente almacena el número de alófono que ha de sonar, comienza la ejecución. Sin embargo, ese sonido continuará produciéndose si no ponemos nueva información en el registro. Poe eso es importante que después de acabar una palabra o frase añadamos una de las cuatro pausas posibles (alófonos 0 a 4).

El circuito integrado **SPO256** precisa una referencia de tiempos para hacer correctamente su trabajo, esto queda resuelto con las patillas 27 y 28, etiquetadas como OSC1 y OSC2. Tras ellas hay un circuito oscilador que precisa de un cristal de cuarzo y un par de condensadores. La frecuencia del cristal es la de trabajo. El fabricante del *chip* recomienda que la frecuencia sea de 3,12 MHz, pero hemos probado con cristales de frecuencias en la gama de 3 MHz y el funcionamiento fue perfecto. Lo único que varía es el tono del sonido final. En las tiendas de componentes suelen encontrarse cristales en esta gama de frecuencias.

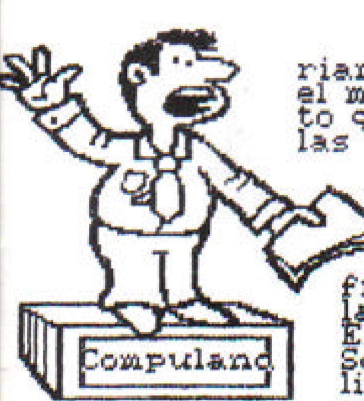
También se podría utilizar una combinación de bobina y condensadores o un oscilador externo.

En las ilustraciones aparecen los elementos necesarios para construir el sintetizador sin mayores problemas. Asimismo proporcionamos listados de ejemplo para que el circuito produzca resultados.

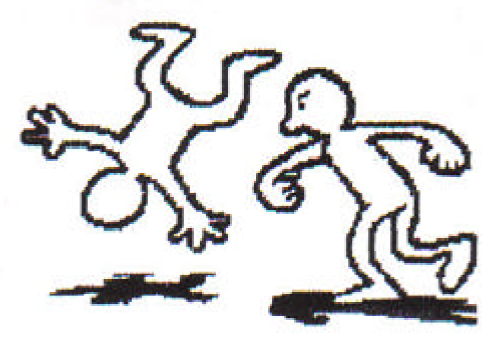
Teclea para Commodore-64

```
 1 REM**DEMOSTRACION SONIDO
   DE LOS ALOFONOS
 2 PA=56576:REM**PORT A
 3 PB=56577:REM**PORT B
 4 DA=56578:REM**DDR A
 5 DB=56579:REM**DDR B
20 FOR X=64 TO 128:REM**
   CODIGO NUMERICO PARA LOS
   64 ALOFONOS
25 REM**PONE TODAS LAS
   LINEAS DEL PORT A EN
   SALIDA DE DATOS
30 POKE DA,255
35 REM**LINEA 7 DEL PORT B
   (PATILLA L=SBY) EN MODO
   ENTRADA
40 POKE DB,127
```





Amigos Commodorianos:una vez roto el muro del anonimato quisieramos daros las gracias por el interes,simpatia y paciencia que habeis demostrado. Os pido disculpas por el retraso sufrido en el envio de la lista de precios. El tema del Club de Software esta paralizado momentaneamente,en espera de hallar unos mejores enfoques que sean beneficiosos para todos los usuarios y no un grupo de desaprensivos.(Admitimos sugerencias)



Gran Lucha de Precios en Disco Virgen

Despues de la dura batalla librada para poderos ofrecer los discos que ya muchos de vosotros conoceis al increible precio de 290 los normales y 315 los de color, hemos conseguido un nuevo reto,discos de doble cara y doble densidad a 350.Como podeis observar seguimos en la brecha intentando dar el mejor servicio.



CINTA

Como aseguraba he conseguido algunos libros del 128:
C 128 INTERNALS
C 128 TRICKS & TIPS,TODO SOBRE EL 128 y otros en preparacion.

En cuanto a Software,aparte de todas las novedades,tanto nacionales,como internacionales,hay un grupo de Superjuegos,que combinan palanca y estrategia,como :

JET-EIDOLON-SILENT SERVICE-BROAD SIDE-TSUSHIMA-GUNSHIP-DECISION IN THE DESERT y muchos otros mas.

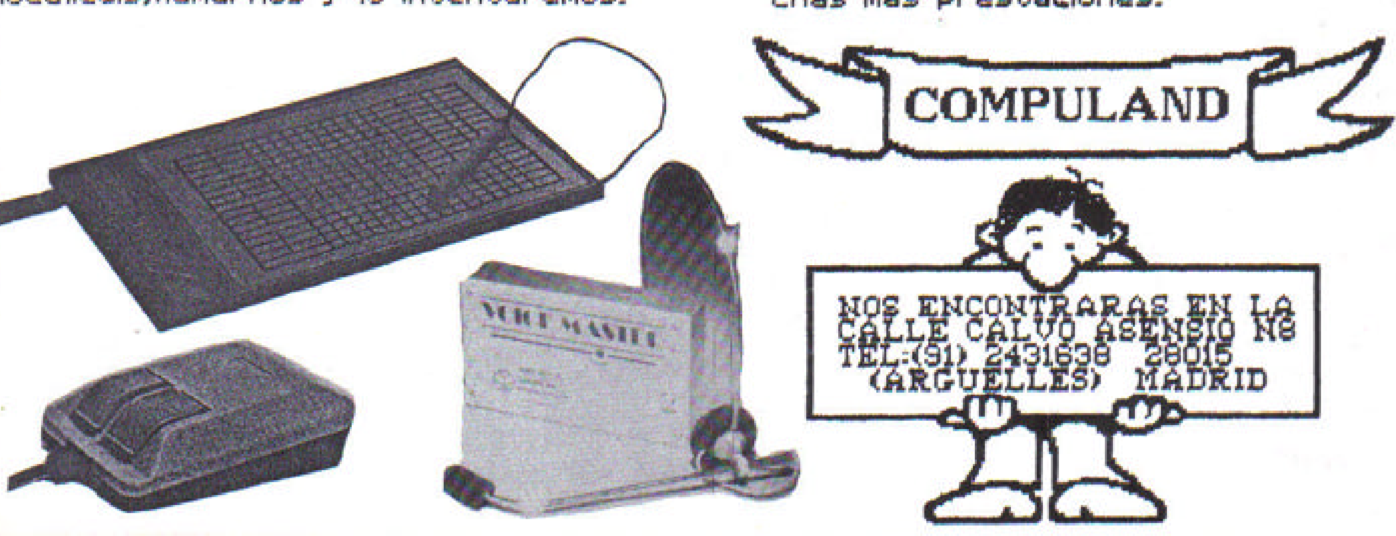
Cualquier articulo de Commodore que os de problemas el poder obtenerlo, o no lo localiceis,llamarnos y lo intentaremos.

NO TE ENFADES

Ni con nosotros,ni contigo.Ya te lo avisamos"CORRE A COMPULAND",pero lo tomaisteis con calma y sucedio lo predecible,se agotaron los lapices opticos,los ratones,las pinzas y otros articulos.

Ya estan otra vez en stock,los maravillosos lapices,el increible raton(con posibilidad de girar las pantallas,scroll,espejo,zoom,salida impresora,copias y un largo etc..),las tan buscadas pinzas y alguna novedad mas,como puede ser la mejor tableta grafica creada para Co.

Dimensiones 350*260*12mms(A4),Alta resolucion,C.A.D.,1280 * 1024 pixels y muchas mas prestaciones.



COMPULAND

NOS ENCONTRARAS EN LA CALLE CALVO ASENSIO N8
TEL:(91) 2431638 28015
(ARGUELLES) MADRID

```
45 REM**PONE ALD EN 1 LOGICO
   (PATILLA M=BIT 2)
50 POKE PA,255
55 REM**RESET DEL CHIP
60 POKE PB,0
65 REM**PONE A 0 EL BIT 2
   DEL PORT A,PARA MEMORIZAR
   LA DIRECCION EN A1/A5
70 POKE PA,251
75 REM**PONE EN PORT B LA
   DIRECCION DEL ALOFONO
80 POKE PB,X
90 FOR S=0 TO 400:NEXT S
100 NEXT
```

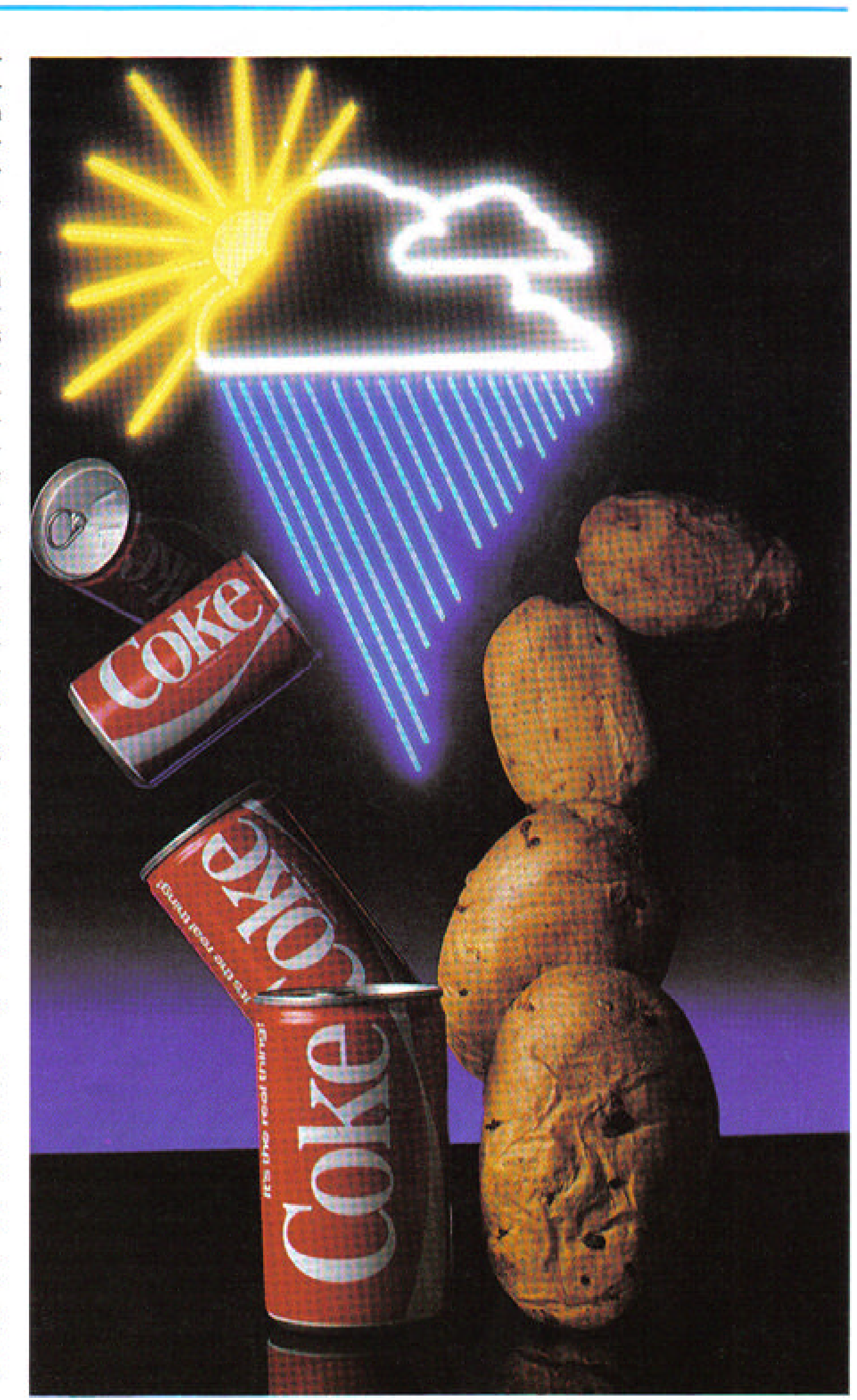
```
1 REM**INPUT
2 PA=56576:REM**PORT A
3 PB=56577:REM**PORT B
4 DA=56578:REM**DDR A
5 DB=56579:REM**DDR B
10 POKE DA,255
20 POKE DB,127
30 POKE PA,255
40 POKE PB,0
50 READ L:IF L=100 GOTO 150
   :REM**EL 100 ES LA CLAVE
   PARA FINALIZAR
60 REM**PONE EN PORT B LA
   DIRECCION DEL ALOFONO
70 POKE PB,X+64
80 REM**NOTIFICAR 'LISTA'
   CARGA DE DIRECCION
90 POKE PA,251
100 POKE 56576,255
110 REM**ESPERAR QUE STY
    (PB7) SEA UN 1 LOGICO
120 IF PEEK(56577)<128 GOTO
    110
130 DATA 12,11,9,30,13,4,100
140 GOTO 50
150 END
```

```
1 REM**BUSQUEDA DE ALOFONOS
2 PA=56576:REM**PORT A
3 PB=56577:REM**PORT B
4 DA=56578:REM**DDR A
5 DB=56579:REM**DDR B
10 DIM A$(64)
20 FOR X=0 TO 63
30 READ A$(X)
40 NEXT
50 DATA PA1,PA2,PA3,PA4,PA5
   ,OY,AY,EH,KK3,PP,JH,NN1,
   IH,TT2,RR1,AX,MM,TT1,DH1
60 DATA IY,EY,DD1,UW1,AO,AA
   ,YY2,AE,HH1,BB1,TH,UH,
   UW2,AW,DD2,GG3,VV,GG1,SH
70 DATA ZH,RR2,FF,KK2,KK1,
   ZZ,NG,LL,WW,XR,WH,YY1,CH
   ,ER1,ER2,OW,DH2,SS,NN2,
   HH2
80 DATA OR,AR,YR,GG2,EL,BB2
90 PRINT"LETRA ?"
100 GET X$:IF X$=""THEN
    100:REM**ALOFONO
    BUSCADO
110 IF X$="/"THEN END:REM**
    TECLEA / PARA TERMINAR
120 FOR X=0 TO 63
130 S$=LEFT$(A$(X),1)
140 IF S$=X$ THEN PRINT X"="
    A$(X)
150 NEXT
160 POKE DA,255:POKE DB,127:
    POKE PA,255:POKE PB,0
170 INPUT"CUAL ELIGES?";AL:
    REM**COMPROBACION SONIDO
180 POKE PB,AL+64:POKE PA,
    251:POKE PA,255
190 IF PEEK(PB)<128 THEN 190
200  GOTO 90
```

SIMULACION: ALIMENTO PARA LA MENTE

- SIMULACION DEL CLIMA
- VARIABLES NORMALMENTE DISTRIBUIDAS
- CONDICIONES COMERCIALES
- TENEDURIA DE LIBROS

Anímate a escribir programas utilizando un método de creación de modelos eficaz, comprobando cómo pueden aplicarse provechosamente los principios del mundo de los negocios a la gestión de un puesto callejero de comidas.



La creación de modelos es un aspecto de gran importancia de la ciencia matemática, sobre el que se han vertido ríos de tinta. Aún hoy en día, las teorías matemáticas no suelen alcanzar el favor del público, especialmente si no pasan del terreno de las hipótesis y no se traducen en algo de aplicación práctica. Pero también en este campo, el advenimiento de los microordenadores ha modificado la tradicional actitud pasiva frente a los problemas matemáticos. De hecho, la potencia de cálculo, presente incluso en el más modesto de los microordenadores, ha eliminado uno de los principales obstáculos psicológicos en la aplicación de las leyes y teoremas matemáticos: lo tedioso de unos cálculos manuales, repetitivos, laboriosos y fácilmente sujetos a errores. En cambio, pensemos en lo fácil que resulta aplicar con un ordenador, por ejemplo, algunas leyes físicas. El movimiento de un péndulo o la caída de un objeto bajo la atracción gravitatoria son experimentos clásicos, pero que implican el uso de elementos e intrumentos de medida no siempre disponibles.

Si disponemos de un ordenador con capacidades gráficas, es muy sencillo verificar dichas leyes: basta con expresar correctamente las fórmulas y utilizar una visualización gráfica lo más parecida posible al caso real. Observar como se comporta el modelo creado en el ordenador, al variara las condiciones que influyen sobre él, es casi un juego (como inciso, muchos de los recientes juegos de ordenador están basados en conceptos similares, eso sí, debidamente camuflados).

La posibilidad de reproducir en el ordenador experimentos que de otro modo resultarían imposibles o peligrosos, tiene como efecto secundario, un desarrollo de la capacidad de síntesis mental. Poder dedicarse totalmente al examen del fenómeno, sin perderse en complejos cálculos, hace más fácil captar la esencia del experimento y, a largo plazo, habitúa la mente a distinguir los factores irrelevantes de aquellos que realmente influyen en un evento determinado.

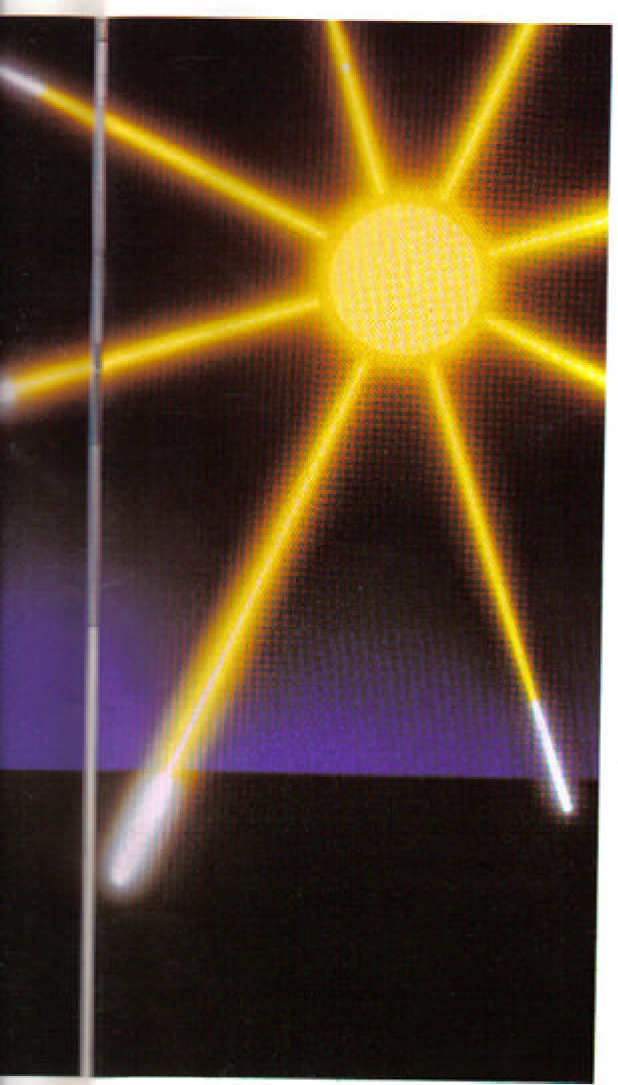
Además, la introducción de elementos aleatorios en un modelo de ordenador, resulta sumamente fácil. Sólo habrá que tener en cuenta la forma de generar diferentes tipos de variables aleatorias que se adapten a los distintos sucesos.

El programa cuyo listado presentamos a continuación, utiliza un método eficiente para generar variables con distribución normal. Te ayudará a crear modelos para un pequeño negocio.

Puedes considerar este programa como un juego (bastante interesante, por cierto), pero no pierdas de vista que, en realidad, se trata de un modelo para una situación real.




```
10 PRINT"[SHIFT+CLR/HOME]
   [CRSR ABAJO]>[CRSR DCHA]
   [CTRL+9]CHIRINGUITO[2*
   CRSR ABAJO]"
20 DIM D$(10)
30 A$(1)="CALIDO Y SECO":D(
   3)=150:D(4)=300
40 A$(2)="CALIDO Y HUMEDO":D
   (5)=100:D(6)=200
50 A$(3)="FRIO Y SECO":D(7)=
   250:D(8)=160
60 A$(4)="FRIO Y HUMEDO":D(
   9)=200:D(10)=100
580 C1(1)=10:C1(2)=20
590 C2(1)=50:C2(2)=25
600 C3(1)=1:C3(2)=12
620 PRINT"CUANTOS JUGADORES
    (1-6)?"
625 GET N$:N=VAL(N$):IF N<1
    OR N>6 THEN 625
630 FOR I=1 TO 2:FOR J=1
    TO N
640 TP(I,J)=0
650 NEXT J,I
700 FOR K=1 TO 10
710 P1=.1*INT(10*(.3+RND(1)/
    2))
720 P2=.1*INT(10*(.2+RND(1)/
    2))
730 PRINT"[SHIFT+CLR/HOME]
    [CRSR ABAJO]>[CRSR DCHA]
    [CTRL+9]DIA",K"[2*CRSR
    ABAJO]"
740 PRINT"PROBABILIDAD DE
    DIA":PRINT"CALIDO ES";
    100*P1;"%"
750 PRINT"PROBABILIDAD DE
    DIA": PRINT"SECO ES";
    100*P2;"%"
760 GOSUB 1400
770 U1=RND(1):U2=RND(1)
780 V1=SQR(2*LOG(1/U1))
790 V2=COS(2*[PI]*U2):V3=SIN
    (2*[PI]*U2)
800 Z1=INT(V1*V2):Z2=INT(V1*
    V3)
810 A1=P1*P2:A2=P1
820 A3=P1+P2-A1:A4=1:F=RND
    (1)
821 IF F<=A1 THEN R=1
822 IF F>A1 AND F<=A2 THEN
    R=2
823 IF F>A2 AND F<=A3 THEN
    R=3
824 IF F>A3 AND F<=A4 THEN
    R=4
830 PRINT"EL TIEMPO ES ";A
    $(R)
840 D(1)=INT(D(1+R*2)+Z1*25)
    :D(2)=INT(D(2+R*2)+Z2*
    40)
850 PRINT"LA DEMANDA DE
    PATATAS":PRINT"ASADAS ES
    ";D(1)
```



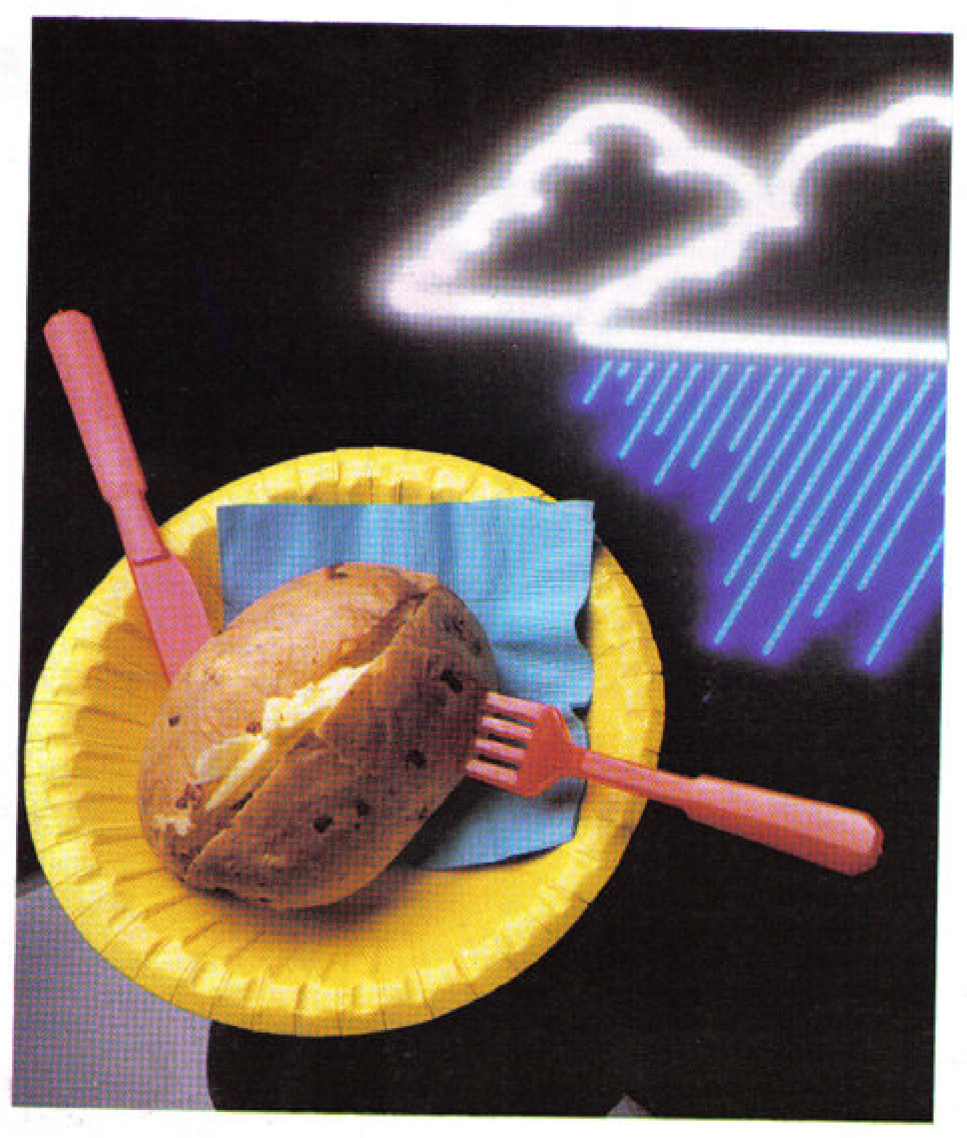
```
860 PRINT"LA DEMANDA DE
    BOTES":PRINT"DE COLA ES"
    ;D(2)
1010 GOSUB 1600
1020 NEXT K
1030 FOR I=1 TO 2000:NEXT I
1100 PRINT"[SHIFT+CLR/HOME]
     RESULTADOS FINALES":
     PRINT"DESPUES DE 10
     DIAS[CRSR ABAJO]"
1110 PRINT"[CTRL+9]JUGADOR
     [CTRL+0][3*CRSR DCHA]
     [CTRL+9]TOTAL
     GANANCIAS"
1120 FOR J=1 TO N
1130 PRINT J,TT(J):NEXT J
1150 END
1400 PRINT"[CRSR ABAJO]>
     [CRSR DCHA][CTRL+9]
     ORDENES POR FAVOR[2*
     CRSR ABAJO]"
1410 FOR J=1 TO N
1420 PRINT"JUGADOR ";J
1430 PRINT"[CRSR ABAJO]
     NUMERO REQUERIDO DE
     PATATAS CALIENTES":
     INPUT O(1,J)
1440 PRINT"NUMERO DE BOTES
     DE COLA REQUERIDOS":
     INPUT O(2,J)
1450 NEXT J:PRINT"[SHIFT+CLR
     /HOME]"
1460 RETURN
1600 FOR J=1 TO N
1610 FOR I=1 TO 2
1620 L=O(I,J)
1630 IF D(I)<L THEN L=D(I)
1650 RV(I)=C2(I)*L
1670 TC(I)=C1(I)*O(I,J)
1680 IF D(I)<=L THEN TC(I)=
     TC(I)-C3(I)*(O(I,J)-D
     (I))
1700 P(I)=RV(I)-TC(I)
1710 TP(I,J)=TP(I,J)+P(I)
1720 NEXT I
1730 TT(J)=TP(1,J)+TP(2,J)
     -2000
1740 E=RV(1)+RV(2)
1750 C=TC(1)+TC(2)+200
1760 P=P(1)+P(2)-200
1770 PRINT"[CRSR ABAJO][CTRL
     +9]JUGADOR "J"[CTRL+0]"
1771 PRINT"[CTRL+9]VENDIDO
     [CTRL+0]  ="E
1772 PRINT"[CTRL+9]COSTES
     [CTRL+0]   ="C
1773 PRINT"[CTRL+9]BENEFICIO
     [CTRL+0]="P
1780 NEXT J
1785 POKE 198,0:WAIT 198,1:
     POKE 198,0
1790 RETURN
```

El programa se concentra en el aspecto de «nadar o hundirse» propio de un negocio: la pérdida o la ganancia. Al teclear RUN para ejecutarlo, puedes elegir entre ser el único participante o actuar junto a otros empresarios, hasta un máximo de cinco. También puedes elegir el jugar la parte de dos o incluso tres de los empresarios, tomando decisiones distintas en cada uno de los tres papeles. Puedes aprovechar la oportunidad para comparar los resultados de, por ejemplo, llevar los negocios con cautela en uno de los casos, y actuar con carácter más emprendedor y asumiendo riesgos en el otro. Sea la que fuere tu elección, al empezar la ejecución del programa debes introducir el número de jugadores.

Cada jugador regenta un puesto que vende patatas cocidas y refrescos de cola. La demanda depende del tiempo que haga. Cuando hace frío, las patatas se venden bien. Cuando luce el sol, se venden muy bien las bebidas refrescantes. Por desgracia el empresario tiene que comprar un stock la víspera de cada día que abre su quiosco, y no conoce el tiempo que va a hacer. Afortunadamente dispone del pronóstico del hombre del tiempo, que suele ser correcto el 70 por ciento de los días. El ganador del juego es el jugador que después de 10 días de comprar y vender haya hecho las mayores ganancias.

La primera parte del programa (hasta la línea 650) establece las variables para la representación en pantalla y las condiciones del negocio. Puedes pagar una renta de 200 pesetas por día. Las patatas las compras a 10 pesetas cada una y las vendes a 50 pesetas. La Coca-Cola te cuesta 20 pesetas y la vendes a 25. Cada día de negocio te deshaces de la mercancía que te sobra a un precio de desecho: 1 peseta para las patatas y 12 pesetas para la Coca-Cola. Cada juego dura diez días.

El indicador de demanda del género es el mostrado en la tabla adjunta, y aquí es donde se introduce un importante elemento en el modelo. Como es natural, los mejores días para las patatas calientes son de los peores para las bebidas frías; sin embargo la demanda está lo suficientemente repartida entre los dos artículos como para que los empresarios hagan un razonable negocio, haga el tiempo que haga. No obstante, todo depende en alguna medida del pronóstico del tiempo, no perdiendo de vista en ningún momento que no es totalmente fiable. Las probabilidades de que haga un día caluroso y seco están determinadas por las líneas 710 y 720;

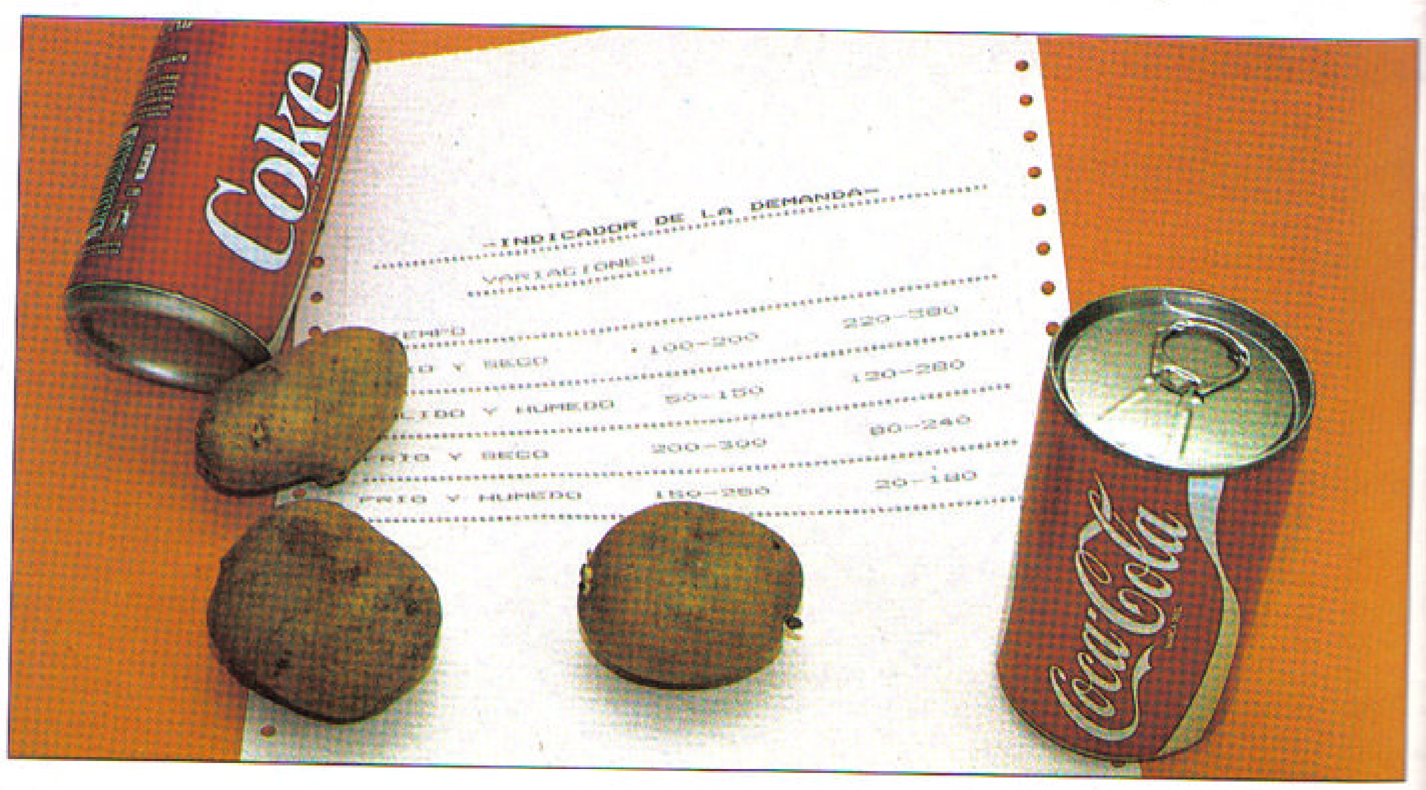


posteriormente se utilizan (líneas 810 a 824) para simular el tiempo.

Las líneas 770 a 800 utilizan un método sofisticado para generar variables aleatorias con distribución normal. Después se usan dichas variables para simular la demanda en la línea 840. La línea 770 genera dos variables aleatorias (U1 y U2), pero recuerda que no son verdaderamente aleatorias. Por ello se procesan en tres fórmulas matemáticas. En la línea 780, se halla el inverso de U1 y se eleva al cuadrado, obteniendo después su logaritmo natural. Finalmente se asigna la raíz cuadrada del resultado como valor de la variable V1. La línea 790 asigna a V2 el coseno de una circunferencia de radio U2, y a V3 el seno del mismo arco. Las variables V1, V2 y V3 sufren una nueva manipulación (línea 800) para dar finalmente las variables normalmente distribuidas Z1 y Z2.

El resto del programa, desde la línea 1600 hasta el final, se dedica a la organización de la entrada de datos y la presentación de resultados.

Cuando juegues este juego tienes que tener en cuenta hasta la última peseta. Los resultados pueden ser atrozmente parecidos, incluso transcurridos diez días de negocio.



SI BUSCAS LO MEJOR



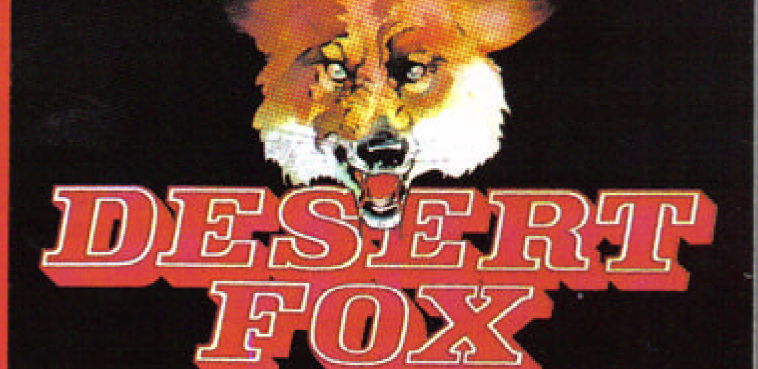
EL MAS COMPLETO SIMULADOR DE GUERRA

UN VIAJE ATRAVES DEL TIEMPO

TIME TUNNEL

Commodore 64/128

A CUNNING FOE WHO KNOWS NO FEAR



CBM64/128



REALISTIC VOICE SIMULATION
Listen in on enemy radio messages, plan your tactics to combat their manoeuvres.

Sydney



TIME TUNNEL

ATREVETE A UN MISTERIOSO VIAJE EN EL TUNEL DEL TIEMPO. DESDE EL REMOTO AÑO 9600 ANTES DE CRISTO HASTA EL FUTURO DEL AÑO 10.000. INTENTA SUPERAR LOS PELIGROS DE LA EDAD DE PIEDRA, EVITA LOS HECHIZOS BRUJOS DEL SALEM DE 1692, CONOCE LA PERSIA MAGICA, ENFRENTATE A MEDUSA EN LA MITOLOGICA GRECIA O VIVE EL VERTIGO DEL VIAJE GALACTICO A TRAVES DEL AGUJERO NEGRO.
TIME TUNNEL ES EL DESAFIO A TU IMAGINACION

DESER FOX

¡LA GUERRA HA ESTALLADO EN AFRICA! LAS CONDICIONES DE LUCHA SON INFRAHUMANAS. EL CALOR AGOBIANTE Y EL MARISCAL ROMMEL HACEN IMPOSIBLE LA SUPERVIVENCIA DEL EJERCITO ALIADO. TU ERES UN LOBO SOLITARIO TU ENEMIGO ES EL ZORRO DEL DESIERTO.
¡LA BATALLA COMIENZA!

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE C/. STA. ENGRACIA, 17. 28010 MADRID, TFNO.: (91) 447 34 10
DELEGACION BARCELONA, AVDA. MISTRAL, N.º 10 - TFNO.: (93) 432 07 31

Programación

UTILIDADES DE EDICION EN DISCO

- ACCESO DIRECTO
- LECTURA Y ESCRITURA
- EDICION EN DISCO
- REIMPLANTACION DE UN FICHERO DESTRUIDO

Para quienes utilizan *diskettes*, existe una utilidad que permite el acceso directo a la información almacenada, haciendo posible corregirla o recuperarla en caso de accidente.

La lectura y escritura de ficheros en un disco están controladas por sistema operativo en disco (DOS) o sistema de ficheros en disco (DFS) que utilice tu ordenador. En condiciones normales, no tienes que ocuparte en absoluto de la organización de la información en el disco, ni de la transferencia de datos al disco o desde el disco, ya que el DOS se ocupa de todo. Los discos actúan simplemente como una extensión del ordenador, un dispositivo de almacenamiento de datos.

Pero el DOS te brinda la posibilidad de acceder a zonas del disco, de forma que se pueda extraer, manipular o corregir información, lo cual constituye una cualidad muy útil que te abre nuevas e interesantes posibilidades.

Puedes utilizar esta forma de acceso directo para cambiar elementos del directorio, nombres de ficheros, ficheros de datos, relaciones entre ficheros, o para recuperar información. Esto último es quizá lo más útil. El acceso directo te permite —entre otras cosas— «arreglar» ficheros que han sido destruídos o arañados, cerrar ficheros que quedaron abiertos o establecer nuevos punteros sectoriales que han resultado deteriorados, restaurando así el correcto «encadenamiento» de los sectores que integran un fichero particular. Hay otros usos que hacen de un programa monitor de disco una utilidad muy interesante.

El acceso directo a «bloques» de información individual en un disco puede asimilarse a la utilización de un **monitor** para examinar y modificar determinadas partes de la memoria. De hecho el siguiente programa se parece mucho en su utilización a uno de tales monitores.

La esencia de un **monitor** de disco (o editor de disco) es su operación por medio de un *buffer* que permite el almacenamiento temporal de los datos leídos en el disco o escritos en el mismo. Mientras la información está en el *buffer*, puede modificarse a voluntad, y esta nueva información puede volver a escribirse encima de la antigua sobre el disco.

GEOGRAFIA DEL DISCO

Si tienes que localizar pista y sectores específicos dentro de un disco, es esencial que tengas algunos conocimientos acerca de la organización de los datos sobre el mismo. Esta organización de datos es lo que normalmente se llama formato del disco, siendo realizada por las rutinas de formateo del DOS.

En todas las referencias relacionadas con el acceso directo se emplea extensamente la notación hexadecimal, por lo que haremos lo mismo en las descripciones generales que siguen. Observa además que la notación hexadecimal no debe ser utilizada en relación con el uso del **DMON** y otros programas de este tipo. Te resultarán por ello de gran utilidad las tablas de conversión hexadecimal y hexadecimal-ASCII.

MANOS A LA OBRA

Antes de poder utilizar el monitor de disco necesitas un mapa del formateado del disco, del cual nos ocuparemos enseguida. Pero antes un par de advertencias. Si estás trabajando con un disco que contiene información importante, haz primero una copia de seguridad (*backup*). De esta forma los posibles fallos que te ocurran no resultarán catastróficos.

En segundo lugar, aunque te parezca obvio, recuerda que no puedes recuperar una información que ya no está ahí. Si por ejemplo quieres recuperar un fichero que se te ha «perdido» o «estropeado», sólo podrás hacerlo si no se ha escrito encima del mismo. Si un fichero ha sido dañado, el espacio de almacenamiento utiliza-



do en una ocasión por ese fichero se deja para futuros usos y podría resultar contaminado si se han escrito otros ficheros mientras tanto.

No obstante, puede ser posible salvar algunos datos dirigiendo los punteros del fichero a las secciones sobre las que no se ha escrito todavía.

La unidad de discos destinada al **Commodore 64** (y al **Vic 20**) es la **1541**. Lleva incorporado su propio sistema operativo en disco, por lo que puede considerarse por derecho propio como un dispositivo «inteligente». Aunque el monitor de disco se puede adaptar para ser utilizado con otras unidades de discos **CBM**, la siguiente información sobre pistas y sectores se aplica específicamente a esta unidad.



Los discos **1541** están divididos en 35 pistas, cada una de las cuales contiene entre 17 y 21 sectores, dependiendo de la situación física de la pista. Hay en total 683 sectores, de los cuales se dispone como máximo de 664. Cada sector tiene espacio de almacenamiento para 256 bytes.

La pista 18 está ocupada por el llamado **directorio**. Normalmente se accede a él tecleando simplemente LOAD«$»,8 seguido de LIST, apareciendo los ficheros secuenciales de programa y de otro tipo que haya en el disco. El directorio puede contener hasta 144 ficheros distintos en un solo disco.

La pista 18 suele ser a la que más frecuencia se accede indirectamente cuando se usa un monitor de disco, para corregir los diversos errores del directorio y otros fallos que puedan ocurrir. Una típica operación de «salvamento de ficheros» es recuperar ficheros que se descartaron por accidente y redefinir los punteros para evitar las zonas deterioradas del disco, que son insalvables.

Pero naturalmente, para orientarte necesitas un «mapa» o formato que te diga dónde y cómo está dispuesta la información en el disco. Fijemos primeramente nuestra atención en la pista 18, el directorio.

En primer lugar está el mapa de disponibilidad de bloques (BAM), cuyo objeto es indicar qué bloques sectoriales pueden utilizarse libremente. El BAM se actualiza después de todo acceso al disco.

La disposición general del BAM y la pista del directorio es la siguiente:

PISTA 18 ($12) SECTOR 0

Bytes $00-$01
Pista y sector del primer bloque del directorio.

Byte $02
Tiene el valor $41 (el carácter A en ASCII, para indicar formato 1541)

Byte $03
Indicador de cero (no es importante)

Bytes $04-$8F
Mapa de bits de los bloques libres (marcados con 1) y los bloques ocupados (marcados con 0)

Bytes $90-$FF
ENCABEZAMIENTO DEL DIRECTORIO

Formato de encabezamiento:

Bytes $90-$A1
Título del disco (separado con espacios, -$A0)

Bytes $A2-$A3
Marca de ID

Byte $A4
Espaciador (espacio $A0)

Bytes $A5-$A6
Tipo de formato ($32, $41 -2A en ASCII)

Bytes $A7-$AA
Espaciadores (espacios $A0)

Bytes $AB-$FF
No utilizado ($00) excepto para la leyenda de BLOQUES LIBRES

Realmente el directorio empieza en la pista 18, sector 1, y según el número de ficheros, puede extenderse también a otros sectores. Todo el resto de la pista 18 se emplea para este fin. En cada sector se pueden especificar un máximo de ocho ficheros.

PISTA 18 ($12) SECTOR 1

Bytes $00-$01
Pista y sector del siguiente bloque del directorio

Bytes $02-$1F
DETALLES DEL PRIMER FICHERO:

Formato del fichero:

Byte $02
Tipo de fichero (ver más adelante)

Bytes $03-$04
Pista y sector del primer bloque de fichero de datos

Bytes $05-14
Nombre del primer fichero (separado con espacios $A0)

Bytes $15-$16
Sólo se utiliza para ficheros relativos (pista y sector de bloques de la cara 1)

Byte $17
Longitud de registro de fichero relativo

Byte $18-$1B
No utilizado

Bytes $1C-$1D
Utilizado para la pista y sector de un fichero nuevo con la instrucción «guardar y sustituir»

Bytes $1E-$1F
Número de bloques del primer fichero (byte bajo, byte alto)

Bytes $20-$21
Espaciador (espacios -$A0)

Bytes $22-$3F
DETALLES DEL SEGUNDO FICHERO (formato de fichero como en el caso del primer fichero)

Los detalles del segundo fichero van seguidos por otro espaciador de dos bytes, y el formato continúa así a razón de ocho ficheros por bloque. Si en un determinado bloque ya no hay más ficheros, el espacio restante se llena con ceros.

Los dos bytes inmediatamente anteriores al nombre del fichero son los punteros del primer bloque de datos de ese fichero particular. Para el primer fichero se encuentran en $03 y $04. Sus valores (en hexadecimal) te dan la pista y el sector. Por ejemplo, $11 $01 indicarían a la pista 17, sector 1. El formato de un bloque de datos típico o de un sector (de 256 bytes) empieza simplemente con el puntero de datos del siguiente bloque en ese mismo fichero, siempre en el orden pista/sector. Los bytes restantes se llenan con datos. El último bloque utilizado por el fichero empieza en $00 y el siguiente valor de byte indica el número de bytes utilizados en ese bloque.

Teclea para Commodore 64

```
10 PRINT"[SHIFT+CLR/HOME]
   [CTRL+8]":POKE 53280,0:
   POKE 53281,0:GOSUB 2400
20 DIM A(255),S(35):HX$=
   "0123456789ABCDEF":SE=1:
   TR=18
30 FOR I=1 TO 17:S(I)=21:
   NEXT
40 FOR I=18 TO 24:S(I)=19:
   NEXT
50 FOR I=25 TO 30:S(I)=18:
   NEXT
60 FOR I=31 TO 35:S(I)=17:
   NEXT
70 PRINT:GOSUB 1000:PRINT" "
80 A$=LEFT$(T$,1)
90 IF A$="I" THEN GOSUB 1200
   :GOTO 70
100 IF A$="X" THEN PRINT
    "BASIC":END
110 IF A$="$" THEN GOSUB
    1500:GOTO 70
120 IF A$>="0" AND A$<="9"
    THEN GOSUB 1600:GOTO 70
130 IF A$="D" THEN GOSUB 650
    :GOTO 70
140 IF A$="S" THEN GOSUB
    1700:GOTO 70
150 IF A$="E" THEN GOSUB
    1800:GOTO 70
170 IF A$="L" THEN GOSUB
    1900:GOTO 70
180 IF A$="ES" THEN GOSUB
    2100:GOTO 70
190 IF A$="C" THEN GOSUB
    2300:GOTO 70
200 IF A$="H" THEN GOSUB
    2400:GOTO 70
210 PRINT".?CD?":GOTO 70
650 OPEN 15,8,15:OPEN8,8,8,
    "#":PRINT#15,"U1:"8;0;
    18;0:CLOSE 15:CLOSE 8
655 OPEN 1,8,2,"$"
660 FOR X=1 TO 141:GET#1,A$:
    NEXT
670 T$(0)="BORRADO":T$(1)=
    "SEQ":T$(2)="PROGRAMA":
    T$(3)="USER"
672 T$(4)="RELATIVO"
680 J=17:GOSUB 940
690 N$=B$
700 J=2
710 GOSUB 940
720 I$=B$
730 GET#1,A$
740 J=2
750 GOSUB 940
760 O$=B$
770 FOR L=1 TO 88
780 GET#1,A$
790 NEXT
800 PRINT"NOMBRE DEL DISCO:
    "N$:PRINT"         ID:"I$:
    PRINT"     OS:"O$"[CRSR
    abajo]"
810 PRINT"LONGITUD","TIPO",
    "NOMBRE[CRSR abajo]"
820 FOR P=1 TO 8
830 GET#1,T$,A$,A$
840 IF T$=""THEN T$=CHR$
    (128)
```

```
850 J=15
860 GOSUB 940
870 N$=B$
880 GET#1,A$,A$,A$,A$,A$,A$,
    A$,A$,A$,L$,H$
890 L=ASC(L$+CHR$(0))+256*
    ASC(H$+CHR$(0)):IF L=0
    THEN 930
900 IF ST THEN CLOSE 1:
    RETURN
910 PRINT L*256,T$(ASC(T$)
    -128),N$
920 IF P<8 THEN GET#1,A$,A$
930 NEXT: GO TO 820
940 B$=""
950 FOR L=0 TO J
960 GET#1,A$
970 IF A$<>CHR$ (96) THEN
    IF A$<>CHR$(160) THEN B$
    =B$+A$
980 NEXT: RETURN
1000 T$="": PRINT ".";
1010 PRINT"[COMMODORE+@]
     [CRSR izquierda]";: GET
     A$: IF A$="" THEN 1010
1020 IF A$=CHR$(13) THEN
     1100
1030 IF A$=CHR$(20) THEN
     1110
1040 IF LEN (T$)>10 THEN
     1010
1050 IF A$=" " OR A$="$"
     THEN 1090
1060 IF A$<"0" THEN 1010
1070 IF A$>"Z" THEN 1010
1090 T$=T$+A$: PRINT A$;:
     GOTO 1010
1100 IF T$<>"" THEN RETURN
1110 IF T$="" THEN 1010
1120 T$=LEFT$(T$,LEN(T$)-1)
1130 PRINT A$;:GO TO 1010
1200 REM RUTINA DE IMPRESION
1210 X$=MID$(T$,3,2):GOSUB
     1300: S=X
1220 X$=MID$(T$,6,2):GOSUB
     1300: F=X
1230 FOR I=S TO F STEP 9
1240 X=I:GOSUB 1400:PRINT
     H$":";:FOR T=0 TO 8:IF
     I+T>255 THEN PRINT"**":
     RETURN
1250 X=A(I+T):GOSUB 1400:
     PRINT H$" ";:NEXT
1260 FOR T=0 TO 8: A=A(I+T):
     IF A<32 OR A>91 THEN
     A=32
1270 PRINT CHR$(A);:NEXT:
     PRINT:NEXT:RETURN
1300 A$=LEFT$(X$,1):B$=
     RIGHT$(X$,1):FOR I=1 TO
     16
1310 IF A$=MID$(HX$,I,1)
     THEN H=(I-1)*16
1320 IF B$=MID$(HX$,I,1)
     THEN L=(I-1)
1330 NEXT: X=H+L: RETURN
1400 H=INT(X/16):L=(X-H*16)
1410 H$=MID$(HX$,H+1,1)+MID$
     (HX$,L+1,1): RETURN
1500 IF LEN (T$)=5 THEN 1540
1505 IF LEN (T$)<>3 THEN
     PRINT ".?SX?";: RETURN
1510 X$=RIGHT$(T$,2)
1520 GOSUB 1300
1530 PRINT ".DEC"X: RETURN
1540 X$=RIGHT$(T$,2):GOSUB
     1300
1550 M=X: X$=MID$(T$,2,2):
     GOSUB 1300:PRINT".DEC"
     256*X+M: RETURN
1600 V=VAL(T$):IF V>65535
     OR V<0 THEN PRINT".??";
     : RETURN
1610 M=INT(V/256)
1620 N=V-M*256
1630 X=M:GOSUB 1400:A$=H$:X=
     N:GOSUB 1400:A$=A$+H$
1640 PRINT".HEX " A$: RETURN
1700 PRINT"ULTIMA PISTA:$";:
     X=TR:GOSUB 1400: PRINT
     H$
1710 PRINT"      SECTOR:$";:
     X=SE:GOSUB 1400: PRINT
     H$
1720 RETURN
1800 X$=MID$(T$,3,2)
1810 GOSUB 1300
1820 A=X:X$=MID$(T$,6,2):
     GOSUB 1300:B=X
1830 A(A)=B:PRINT"OK":
     RETURN
1900 IF LEN(T$)=1 THEN GOSUB
     2000: RETURN
1910 X$=MID$(T$,3,2):GOSUB
     1300:IF X<1 OR X>35
     THEN PRINT".?PI?";:
     RETURN
1920 A=X:X$=MID$(T$,6,2):
     GOSUB 1300:IF X<0 OR X>
     S(A) THEN PRINT ".?SI?"
     ;:RETURN
1930 TR=A:SE=X:GOSUB 2000
1940 RETURN
2000 OPEN 15,8,15
2010 OPEN 8,8,8,"#"
2020 PRINT#15,"U1:"8;0;TR;SE
2030 PRINT#15,"B-P:"8;0
2040 FOR I=0 TO 255:GET#8,A$
2042 IF ST<>0 AND ST<>64
     THEN PRINT".?DL?":CLOSE
     8:CLOSE 15:RETURN
2050 A(I)=ASC(A$+CHR$(0))
2060 NEXT
2070 CLOSE 8:CLOSE 15:PRINT
     "OK": RETURN
2100 IF LEN(T$)=1 THEN GOSUB
     2200: RETURN
2110 X$=MID$(T$,3,2):GOSUB
     1300:IF X<1 OR X>35
     THEN PRINT".?PI?";:
     RETURN
2120 A=X:X$=MID$(T$,6,2):
     GOSUB 1300: IF X<0 OR
     X>S(A) THEN PRINT".?SI
     ?";:RETURN
2130 TR=A:SE=X:GOSUB 2200
2140 RETURN
2200 OPEN 15,8,15
2210 OPEN 8,8,8,"#"
2220 PRINT#15,"B-P:"8;0
2230 FOR I=0 TO 255
2232 PRINT#8,CHR$(A(I));:IF
     ST<>0 AND ST<>64 THEN
     PRINT".?DE?":GOTO 2250
2240 NEXT:PRINT#15,"U2:"8;0;
     TR;SE
2250 CLOSE 8:CLOSE 15
2260 RETURN
2300 OPEN 15,8,15
2310 INPUT#15,A,B$,C,D
2320 PRINT"ERROR NO.:"A
2330 PRINT"      TIPO: "B$
2340 PRINT"    PISTA:"C
2350 PRINT"    SECTOR:"D
2360 CLOSE 15: RETURN
2400 PRINT"[CRSR abajo]
     I XX XX
     - IMPRIME MEMORIA
2410 PRINT"D
     - DIRECTORIO
2420 PRINT"L XX XX
     - LEE DEL DISCO
2430 PRINT"ES XX XX
     - ESCRIBE EN DISCO
2440 PRINT"E XX XX
```

```
     - EDITA MEMORIA
2450 PRINT"S
     - ULTIMO SECTOR/PISTA
2460 PRINT"$
     - HEX A DECIMAL
2470 PRINT"(NUMERO)
     - DECIMAL A HEX
2480 PRINT"C
     - ULTIMO ERROR
2490 PRINT"X
     - SALIR AL BASIC
2500 PRINT"H
     - IMPRIME MENU
2510 PRINT"[CRSR abajo]PI
     - PISTA ILEGAL
2520 PRINT"SI
     - SECTOR ILEGAL
2530 PRINT"SX
     - ERROR DE SINTAXIS
2540 PRINT"CD
     - COMANDO DESCONOCIDO
2550 PRINT"DL
     - ERROR LECT. DISCO
2560 PRINT"DE
     - ERROR ESCRIT. DISCO
2570 RETURN
```

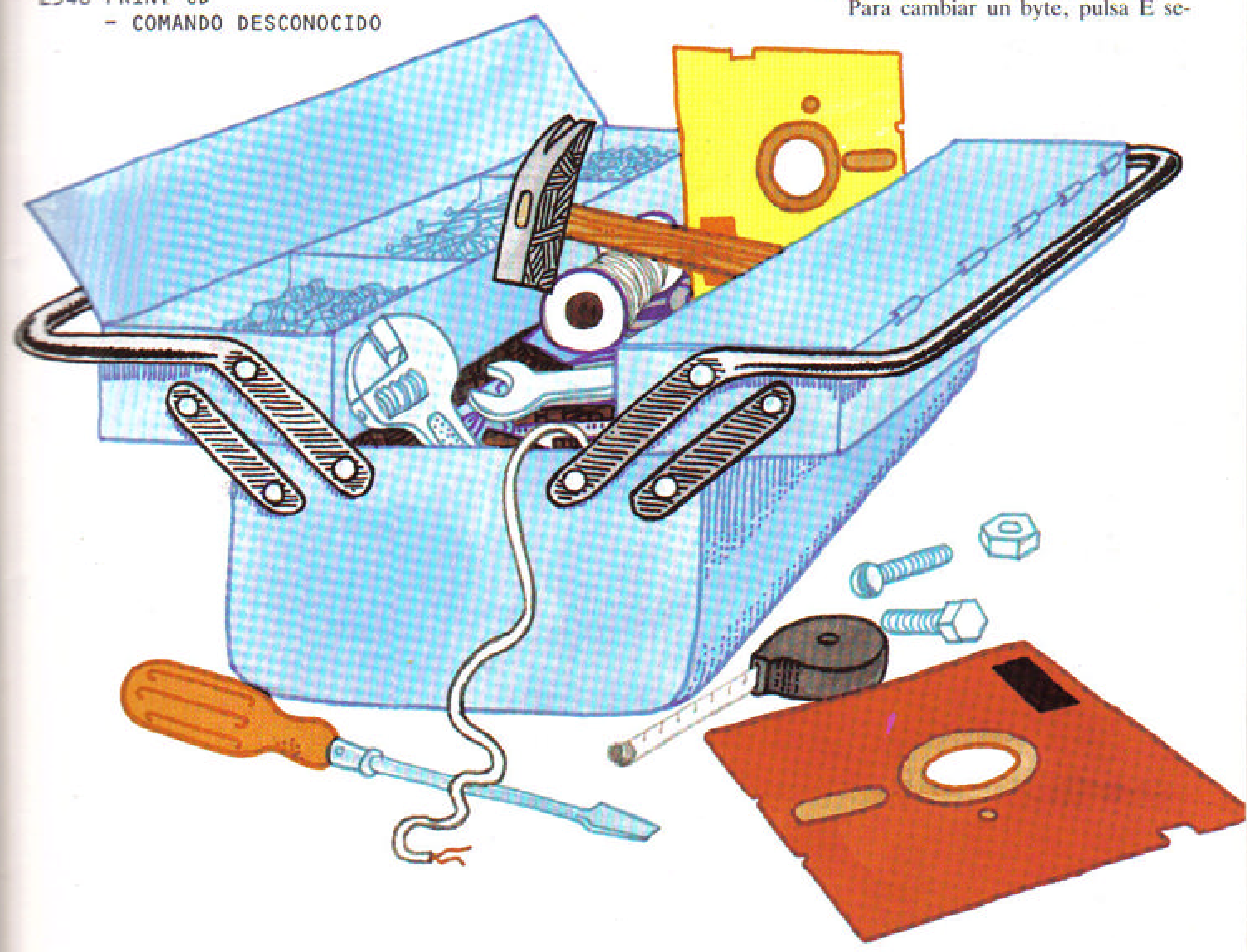
USO DEL DMON

Al iniciar la ejecución del programa (RUN) verás un menú en el que aparecen once comandos, así como una lista de mensajes de error con sus explicaciones. Se accede a cada comando con la letra que se indica, y las X dobles significan que tienes que teclear un número según las instrucciones que vermos más adelante.

Para tener un directorio completo, pulsa D. Las longitudes de los programas se expresan en bytes y se muestran tanto los nombres antiguos de ficheros como los nuevos.

Para leer cualquier parte del disco, pulsa L y a continuación el número de la pista y el sector. Acuérdate de usar números en hexadecimal. Haz una prueba pulsando L 12 01 para leer el directorio tal como aparece en el disco. Los datos leídos son almacenados en el *buffer* de disco y pueden presentarse en pantalla pulsando I, seguido de los números de los bytes primero y último de la sección que deseas ver. Para ver todo el contenido, utiliza I 00 FF. Los números de bytes se muestran en la columna de la izquierda, el contenido hexadecimal en el centro y su equivalente en ASCII a la derecha.

Para cambiar un byte, pulsa E se-



guido del número de byte que quieras cambiar y del nuevo valor. Verás cómo el nuevo valor aparece en el sitio correcto en la pantalla. Para escribir esto sobre el disco, pulsa ES seguido de la pista y sector de destino.

Los restantes comandos de que dispones son S, para presentar la última pista y sector a que se ha accedido; $ seguido de un número en hexadecimal, para convertirlo en decimal; un número decimal, para convertirlo en hexadecimal; C, para obtener el código del último error; H para presentar de nuevo el menú y X para volver al BASIC.

Cuando un fichero es deteriorado, el marcador de tipo que aparece en el directorio resulta alterado. Este marcador de tipo va situado inmediatamente antes de los punteros de pista y sector (P&S) que a su vez van inmediatamente antes del nombre del fichero. En el listado de un directorio pueden aparecer ficheros de varios tipos: SEQ, PROGRAMA, USR y RELATIVO.

Los ficheros son normalmente abiertos —lo que significa que se puede escribir en ellos (encima de lo que hubiera antes)— o cerrados. Los ficheros se abren cuando resultan «arañados» para liberar sus posiciones de bloques de almacenamiento. Cuando un fichero se cierra, se hace «activo» en el sentido de que queda almacenado. Pero es posible bloquear estos ficheros de forma que no puedan dañarse fácilmente.

Con el monitor de disco puedes encontrar con facilidad el byte del tipo de fichero y examinar su *status*. Existen los siguientes tipos de ficheros con sus correspondientes designaciones en hexadecimal:

Tipo de fichero	Cerrado	Abierto	Protegido
Borrado	$80	$00	---
SEQuencial	$81	$01	$C1
PROGRAMA	$82	$02	$C2
USeR	$83	$03	$C3
RELATIVO	$84	$04	$C4

Así, un fichero de programa que ha sido dañado, presentará un 02 en la posición del byte de tipo de fichero. Si no se ha escrito sobre los bloques o sectores de ese fichero, se pueden recuperar los datos con el monitor de disco cambiando el anterior valor por $82 (el valor que tendría el fichero si fuera activo).

Pero si cambias el valor a $C2, puedes realmente bloquear el fichero para impedir que se deteriore. Del mismo modo puedes usar los valores apropiados de la tercera columna para proteger otros tipos de ficheros. Cuando hagas el siguiente listado de directorio, verás que las letras del tipo de fichero van acompañadas de un signo <, para indicar la protección. Si quieres eliminar ficheros, puedes hacerlo con NEW de todo el disco, o utilizando el editor para cambiar los valores adecuadamente a fin de que se «abran».

Si ha resultado deteriorada una parte de un fichero —como sería el caso de un fichero sobre alguno de cuyos bloques se ha escrito un fichero activo— tienes que seguir la pista a los diversos punteros de pista y sector para evaluar la extensión de los daños.

Empieza por el directorio ($12 $01) y define el puntero P&S correspondiente a la primera pista de datos que quieras recuperar. A continuación puedes utilizar el monitor de disco para examinar esa pista. Si dicha pista está intacta, es decir si lo que contiene no es un galimatías indescifrable, sigue examinando la siguiente de la cadena. La correspondiente posición de pista y sector está dada por los dos primeros bytes del sector que estás examinando.

Si llegas a un sector en que se aprecian daños, mira primero si es posible arreglarlo simplemente escribiendo de nuevo uno o dos de los anteriores punteros. Al escribir de nuevo uno de los punteros anteriores el fichero se salta el sector estropeado. Esto te permitirá recuperar la información con el programa matriz, con su información cuidadosamente ordenada de forma que el final de cada sector se enlaza con el principio del siguiente, y volver a archivar de nuevo todo el fichero en un nuevo disco.



PERSONAL PRINTER

RITEMAN C+

BUILT-IN COMMODORE INTERFACE

MANUAL DE INSTRUCCIONES EN CASTELLANO

NOVEDAD: C + versión NLQ

INFORUNNER

RITEMAN C+

ESPECIFICACIONES

1. **Método de impresión**
   Serie de impactos por matriz de 9×9
2. **Cabezal impresor**
   9 agujas (reemplazable) con expectativa de vida de 50 millones de caracteres a 100 % de uso continuo.
3. **Velocidad de impresión**
   Impresión a 105 cps uni o bi-direccional, búsqueda lógica optimizada, capacidad de impresión 45 lpm a 10 cpi., 200 ms de salto de línea con espaciado de 1/6".
4. **Espaciado interlínea**
   1/6", 1/8", 7/72", programable N/72", N/216".
5. **Alimentación de papel**
   a) Tractor ajustable entre 4" y 10".
   b) Alimentación a fricción para hojas sueltas entre 4" y 9,5".
6. **Corte de papel**
   Se puede imprimir desde principio de página hasta una pulgada de fin de papel.
7. **Selección de caracteres**
   96 caracteres ASCII, 96 itálicos, 82 caracteres gráficos Commodore. Definición de caracteres por usuario a través de software fácilmente.
8. **Gráficos**
   Normal 480 puntos por línea. Doble densidad 960.
9. **Modos de impresión**
   Modo Commodore
   Standard, reverse (negativo), mayúsculas y minúsculas y caracteres gráficos.
   Modo plus
   Standard, doble impresión, enfatizado, itálica, supra y subíndices mitad altura.
10. **Tamaños de impresión**
    Modo Commodore
    a) Normal
    b) Expandido
    c) Expandido doble
    d) Expandido triple
    e) Expandido cuádruple.
    Modo plus
    a) Normal (10 cpi), 2,1 mm (a)×2,55 mm(h).
    b) Normal expandido (5 cpi)
    c) Comprimido (17 cpi)
    d) Comprimido-expandido (8,5 cpi).
11. **Test escritura**
    Manteniendo apretado LF (salto de línea) cuando se acciona el interruptor de puesta en marcha ON.
12. **Subrayado continuo.**
13. **Tabulador**
    Existen 32 posiciones de tabulación por línea.
14. **Longitud de formato**
    El standard de 66 líneas, con espaciado de líneas y longitud del formato programable.
    VFU eléctrico.
15. **Tamaño buffer**
    1 línea standard.
16. **Diagnóstico de detección de errores en modo Commodore.**
17. **Posicionador de papel continuo**
    Desplazable parte anterior o posterior.
18. **Interface**
    Directo a Commodore salida RS232 a través de conector DIN (el cable se suministra con la impresora).
19. **Grosor papel**
    Máximo 2 copias, 40 g/m² peso continuo, sensor papel.
20. **Operación con interruptores**
    De fácil acceso.
21. **Caracteres por línea**
    Normal: 80
    Normal expandido: 40
    Comprimido: 132
    Comprimido expandido: 66
22. **Cinta impresora**
    Autorretintada, mediante tampón interno, cartucho compacto de cinta negra fácilmente reemplazable y bajo coste.
23. **Tamaño**
    115 (h) × 275 (f) × 405 (a).
24. **Tensión alimentación**
    220 v/ 50 Hz.
25. **Entrada de papel**
    Frontal-horizontal.
26. **Retroceso**
    Por software.
27. **Instrucciones de formateado de tablas alfanuméricas**
28. **Elevadores inferiores standard**
    (para ubicación del papel continuo debajo de la impresora).
29. **Motor: paso a paso**
30. **Indicadores:**
    - Conexión general
    - Fin de papel
    - On line
31. **Interruptores**
    ON/OFF, On line (SEL), salto de página (FF), salto de línea (LF).
32. **Compatible con la mayor parte de software existente**
    (incluso Commodore).

DATAMON
DATAMON, S. A.

REPRESENTACION EN ESPAÑA DE


IMPRESORAS PROFESIONALES

- PERIFERICOS ORDENADORES
- IMPRESORAS Y MONITORES
- CONEXIONES Y ACCESORIOS

PROVENZA, 385-387, 6.°, 1.ª
TELEFONO (93) 207 27 04 · 08025 BARCELONA

TODAS LAS ESPECIFICACIONES PUEDEN SER CAMBIADAS SIN PREVIO AVISO. * Commodore es marca registrada de Commodore Bussines Machine, Inc.

ASI ES EL MAPA DE MEMORIA DEL COMMODORE 64

Conocer al detalle como es el mapa de memoria de cualquier ordenador es asunto primordial para todos aquellos que deseen programar en código máquina de manera eficiente. INPUT irá ofreciendo número a número no sólo la conocida lista de direcciones y su denominación, sino un comentario de las posibilidades de cada uno de los 64 kbytes significativos del 64, huyendo de los escuetos comentarios que ofrecen los mapas de memoria habitualmente divulgados.

Byte $0000 = 0

Registro de dirección de los datos, incluido en el 6510. El registro situado en la dirección $0001 actúa como *port* de Entrada/Salida que va incorporado en el propio microprocesador.

Los bits de este registro definen si el correspondiente en la dirección 1 actúa como Entrada (cuando vale 1) o Salida (puesto a 0).

Bit 0 controla la dirección del bit 0 del registro $0001. Al conectar el ordenador es puesto a 1 (actuará como salida).

Bit 1, inicialmente a 1 (salida)
Bit 2, inicialmente a 1 (salida)
Bit 3, inicialmente a 1 (salida)
Bit 4, inicialmente a 0 (salida)
Bit 5, inicialmente a 1 (salida)
Bit 6, no utilizado normalmente.
Bit 7, no utilizado normalmente.

Byte $0001 = 1

Los bits de información depositados en esta dirección proporcionan mucho juego. Cada uno asume una importante misión.

El bit 0 controla uña señal llamada LORAM, pudiendo desconectar por si mismo a toda la ROM que contiene al BASIC, haciendo que sea directamente accesible la RAM situada en las mismas direcciones de memoria y que en condiciones normales es solapada y sustituida por dicha ROM. Las direcciones de memoria afectadas son las que se situan entre $A000 (40960) a $BFFF (49151).

Habitualmente es bit tiene valor 1, el 0 desactiva automáticamente a la ROM.

El bit 1 controla otra señal conocida por HIRAM. Su actuación es equivalente a la del anterior bit, sólo que esta vez es la ROM que contiene al sistema operativo **Kernal** el área de memoria que puede ser desactivada. En tal caso también queda a disposición del programador toda la zona de memoria situada entre las direcciones $E000 (57344) y $FFFF (65535). El valor normalmente atribuido a este bit es 1, la ROM queda desactivada cuando se cambia por un 0.

Estos bits son especialmente relevantes cuando un nuevo programa requiere un sistema operativo propio o en los casos en que se quiere prescindir del BASIC para sustituirlo por un programa en código máquina de aplicación específica o un lenguaje de programación diferente, tal como el PASCAL, el FORTH, o el LISP, por ejemplo.

El bit 2 es responsable de controlar la señal CHAREN, dejando activa o desactivando la ROM que ocupa el área de memoria comprendida entre $D000 (53248) y $DFFF (57343). Como en los dos casos anteriores, un valor de 1 la activa y el 0 se ocupa de lo contrario.

Esta memoria ROM contiene la información relativa a todos los caracteres que pueden aparecer en la pantalla, sabiendo que puntos deben encenderse y cuáles no sobre la rejilla imaginaria de 8 por 8 puntos que utiliza cada caracter. Tomando esta información y volcándola en algún área de la RAM, es fácil alterar la composición de los caracteres para diseñar un conjunto propio.

Existe un inconveniente. Esta misma área de memoria es utilizada por los *chips* que disponen de funciones de Entrada/Salida, como el VIC-II, el SID y las dos CIAs 6526. Sin embargo, el sistema operativo hace una trampa que obliga al *chip* VIC-II a buscar la descripción de los caracteres en dos áreas de memoria, situadas entre $1000 (4096) y $1FFF (8191) por



un lado y $9000 (36864) a $9FFF (40959) por el otro.

El tercer bit está directamente conectado con la patilla de escritura del *port* del *cassette*, actuando como línea de salida de datos hacia la unidad de cinta. Que trabaje como salida fué definido en el bit 3 de la dirección $0000.

El bit 4 trabaja en sentido de entrada de datos, recibiendo información desde la unidad de *cassette* para conocer si la cinta está siendo desplazada por el motor.

El bit 5 es de salida y controla el motor de la unidad de *cassette*. Cuando su valor es 0 el motor está libre para girar y un 1 se lo impide en cualquier circunstancia.

Byte $0002 = 2

Es empleado a guisa de comodín, normalmente no es utilizado.

Byte $0003 = 3
Byte $0004 = 4

Ambos son un vector que apunta hacia una dirección de comienzo de una rutina que transforma un valor de punto (coma en nuestro sistema de numeración) flotante en un número entero (punto fijo).

El byte $0004 es el de más peso y el $0003 el de menos, siguiendo la filosofía empleada por el microprocesador para acceder a las direcciones definibles con 16 bits.

Normalmente el contenido de estos dos bytes es $AA para el primero y $B1 para el segundo, señalando a la mencionada rutina en la dirección $B1AA (45482).

Byte $0005 = 5
Byte $0006 = 6

Este par de bytes apuntan a una nueva dirección que hace justo lo contrario de la anterior, convirtiendo un valor entero en su correspondiente en coma flotante.

La rutina comienza en la dirección $B391 (45969), por lo que el byte 5 contiene el valor $91 y el byte 6 el $B3

Byte $0007 = 7

Este byte tiene una interesante aplicación, que consiste en la búsqueda de determinados caracteres clave que pueden existir en la línea de texto del programa en BASIC que estamos introduciendo en el *buffer* (memoria intermedia de manejo más cómodo para el ordenador en determinadas tareas) destinado al análisis previo a su ejecución o almacenamiento y que comienza en la dirección $0200 (512).

El byte 7 es muy utilizado por las rutinas que exploran dicho texto, buscando caracteres tales como separadores (dos puntos), terminadores (fin de línea de programa), coma o comillas. Esos caracteres son almacenados temporalmente en esta dirección para su posterior uso.



AÑO 1 NUMERO 7

DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Luis R. Palencia, Christophe Pais, Francisco Tórtola, Benito Román, Esther de la Cal.

INPUT Commodore es una publicación juvenil de EDICIONES FORUM

GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: Grupo Jota
Madrid: c/ Cuenca, 1, 1.°
Teléf. 253 45 01/02
Barcelona: Avda. de Sarriá, 11-13, 1.°
Teléf. 250 23 99

FOTOMECANICA: Ochoa, S. A.

COMPOSICION: EFCA, S. A.

IMPRESION: Edime, S. A.
Depósito legal: M. 27.884-1985

SUSCRIPCIONES: EDISA,
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12

REDACCION:
Alberto Alcocer, 46, 4.°
28016 Madrid. Teléf. 250 10 00

DISTRIBUIDORA
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona
El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

Se ha solicitado el control OJD

INPUT Commodore es independiente y no está vinculada a Commodore Business Machines o sus distribuidores.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.

INPUT commodore

SUMARIO

Byte %0008 = 8

Su utilización es semejante a la del byte anterior. Habitualmente almacena el valor de un terminador o comillas durante la compresión de las **palabras clave** del BASIC. También se utiliza como campo de trabajo para la función AND.

Byte $0009 = 9

Se encarga de memorizar la posición de los caracteres en las rutinas SPC (espacios) y TAB (tabulación). El valor que puede contener se situa entre $00 (0) y $4A (79).

Byte $000A = 10

Controla que la rutina de carga del sistema operativo realice carga (LOAD) o verificación de datos (VERIFY).

Byte $000B = 11

Esta dirección actúa como índice señalador dentro del área del *buffer* en el que se introduce el texto inicial de una línea de programa en BASIC. Una vez que concluye la tarea de comprimir (tokenizar) las **palabras clave** del BASIC contenidas en dicha línea, el valor que alberga esta dirección es el de la longitud total de la línea comprimida.

En otra faceta, esta dirección de memoria es utilizada para cambiar los modos de trabajo de AND y OR.

Igualmente es utilizada por DIM para evaluar la cantidad de espacio necesario para almacenar una matríz o conocer el número de subíndices empleados por una de ellas dada.

Btye $000C = 12

Este byte incorpora una serie de *flags* (banderas de aviso) que anotan la existencia de determinadas características de una matríz, tales como si la matríz ha sido definida, que una variable dada sea matríz o la existencia de dimensionamiento. Estos *flags* son utilizados por las rutinas encargadas de su manejo.

Byte $000D = 13

Aquí se guarda constancia de si el dato que se está procesando es un valor numérico o una cadena de caracteres. Si contiene el valor $FF es una cadena. Un valor $00 indica el caracter estrictamente numérico del dato tratado en ese momento.

Byte $000E = 14

La información que contiene esta dirección de memoria guarda relación con el *flag* del byte anterior. Un valor $00 advierte que se está manejando un número en punto flotante. Por el contrario, si la información contenida es un $80, se trata de un número entero (punto fijo).

LOS OBJETOS DE LA AVENTURA

- DATA PARA LOS OBJETOS
- DESCRIPCIONES LARGAS Y CORTAS
- MAS VERBOS
- TOMANDO Y DEJANDO OBJETOS
- RUTINA DE INVENTARIO

Ha llegado el momento de llenar tu vacío mundo de aventura con objetos. Te enseñaremos cómo puedes incorporar en el programa tu lista de objetos y la forma de manejarlos.

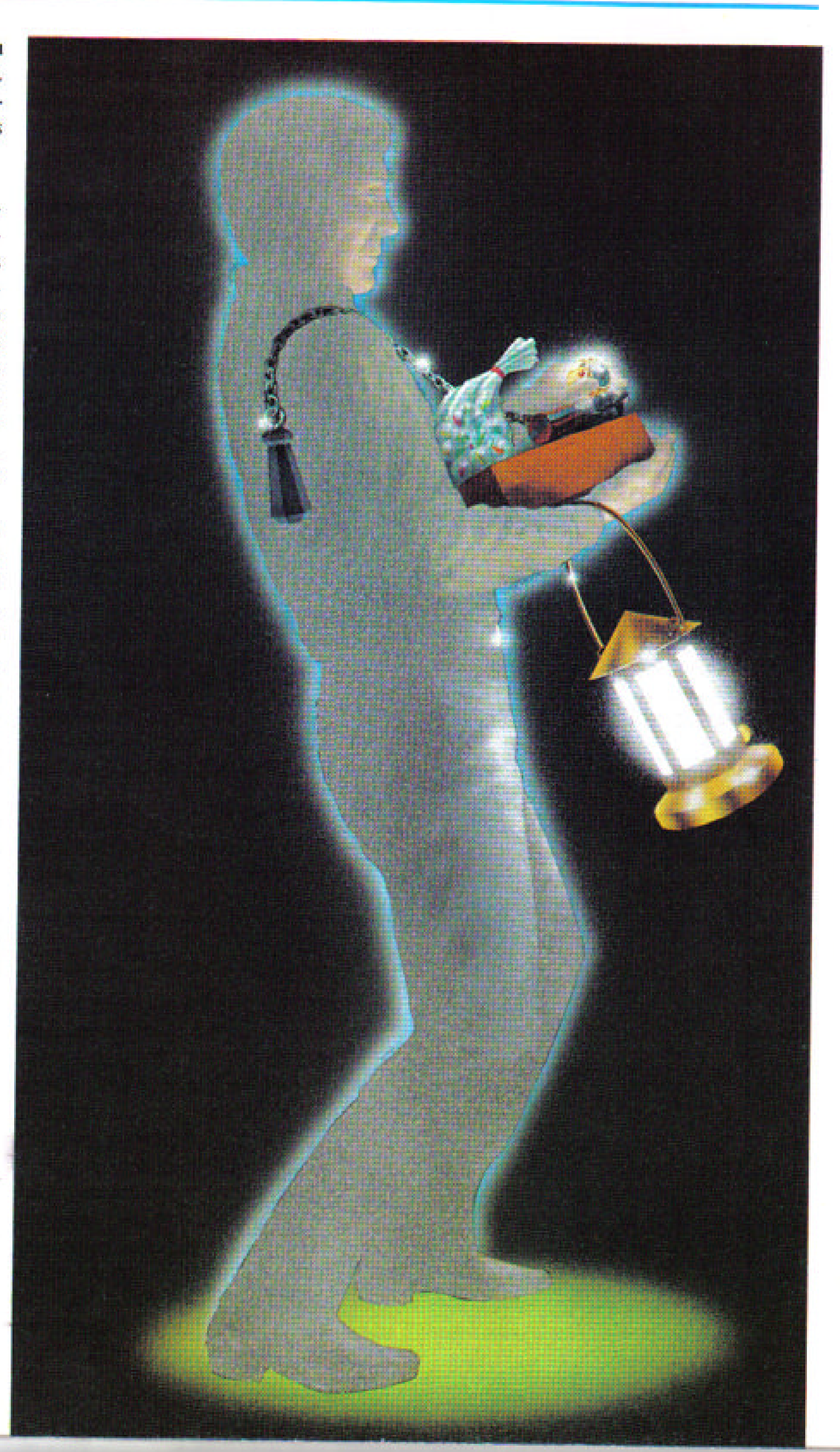
Al final del capítulo anterior teníamos un conjunto completo de ambientes para la aventura y ya habiamos proporcionado al aventurero la capacidad de moverse por todo el mundo de la aventura. Sin embargo en la fase actual las actividades del aventurero todavía carecen de sentido, ya que aún no sucede nada en ninguno de los lugares. Ha llegado el momento de volver atrás y ver lo que habías planeado incluir en cada punto.

A continuación veremos la forma de agregar las rutinas necesarias para llegar al sitio adecuado y recoger o abandonar todos los objetos que intervienen en la aventura. También presentaremos una rutina que hace un inventario de todos los objetos que el aventurero lleva consigo en un momento dado, la cual puede resultar útil en los momentos más críticos de la acción.

Carga el programa que tienes desde el capítulo anterior, dejándolo listo para recibir nuevas rutinas.

OBJETOS

La máquina necesita saber tres cosas acerca de los objetos de una aventura: el número del lugar donde el objeto estaba inicialmente situado, listo para que el aventurero lo encontrara, un nombre para el objeto y una descripción más larga que incluirá algo acerca de la situación del objeto y sugiera de alguna forma su empleo. Estas tres cosas son indispensables para el ordenador, ya que en primer lugar tiene que saber si un determinado objeto está o no en un ambiente, además



tiene que poder avisar al jugador de su presencia, con ayuda de la descripción larga y, por último, necesita un nombre corto, para utilizarlo en instrucciones e inventarios.

Los números de los lugares se situarán en una matriz, el título de los objetos en otro y las descripciones de los objetos en un tercero. El programa procesará las tres matrices en paralelo; cada elemento de la matriz soporta una información equivalente acerca de los objetos, el primero es el número del lugar, el segundo es el nombre del objeto, etc.

Añade a tu programa las siguientes líneas:

Teclea para C-64 y Vic-20

```
160 REM **PREP.MATRICES
    DE LOS OBJETOS
170 READ NB
180 DIM OB(NB),OB$(NB)
    ,SI$ (NB)
190 FOR I=1 TO NB:READ OB
    (I),OB$(I),SI$(I):NEXT
200 DATA 7,4,BOLSA, HAY UNA
    BOLSA DE CANICAS
210 DATA 14,LADRILLO,HAY UN
    LADRILLO A TUS PIES
220 DATA 24,CADENA, UNA
    CADENA CUELGA DEL TECHO
230 DATA 0,PISTOLA, HAY UNA
    PISTOLA EN EL SUELO
240 DATA 0,GLOBO OCULAR,EN
    EL SUELO HAY UNA ALHAJA
    CON FORMA DE OJO
250 DATA 22, LAMPARA, VES
    UNA LAMPARA
260 DATA 0,INSPECTOR FISCAL,
    UN INSPECTOR FISCAL
    APARECE DE REPENTE
```

Cada línea desde la 200 a la 260 contiene tres elementos de los DATA referidos al mismo objeto. La línea 200 incluye un elemento suplementario: el número 7 al principio de los datos informa a la máquina de cuantos conjuntos de datos hay.

Una vez que se lee el número 7 en la línea 170, se dimensionan tres matrices con ese tamaño en la línea 180. OB contendrá la situación de cada objeto, ya sea un número de lugar, un 0 si el objeto todavía no existe (por ejemplo el famoso ojo de brillantes que tiene que ser descubierto durante la aventura), o un −1 si se trata de un objeto que el aventurero lleva consigo. OB$ contendrá las descripciones cortas y SI$ las descripciones largas.

La línea 190 llena las matrices con datos de las líneas 200 a 260. Los datos se disponen en grupos de tres de la forma siguiente: número de lugar, descripción corta del objeto y descripción larga del mismo.

Cuando uses esta rutina para otras



aventuras, no tendrás que hacer muchas modificaciones en su estructura, ya que al ajustar el primer elemento de los DATA, automáticamente se dimensionarán tanto el bucle FOR ... NEXT como las matrices.

DISPOSICION DE LOS OBJETOS

A continuación el programa contiene toda la información referente a la naturaleza y colocación de los objetos.

La siguiente rutina presenta la descripción larga del objeto en el lugar adecuado:

Teclea para C-64 y Vic-20

```
360 REM **VISUALIZACION DE
    LOS OBJETOS EN EL LUGAR
    APROPIADO**
370 FOR I=1 TO NB:IF OB(I)=L
    THEN PRINT SI$(I)
380 NEXT I
```

En esta fase tienes que hacer una pequeña modificación en las líneas 330 y 340: cambia el GOTO 400 por GOTO 370. Las líneas 370 y 380 examinan la matriz que contiene los lugares de los objetos. Si alguno de los números de los lugares coincide con el del lugar donde se encuentra el objeto en ese momento —variable L— aparece la descripción corta del objeto a continuación de la del lugar. Esta rutina se puede usar sin modificaciones en otras aventuras.



MAS VERBOS

Ya tienes en tu aventura unos cuantos objetos esparcidos por los diversos lugares, pero como la máquina todavía no entiende más palabras que NORTE, SUR, ESTE y OESTE, el pobre aventurero no puede hacer nada con esos objetos. Imagínate la frustración de no poder coger esa apetitosa bolsa de canicas o no poder defenderte contra el inspector de hacienda. Por eso tienes que darle al ordenador un vocabulario de palabras que pueda reconocer, diciéndole qué debe hacer con los objetos. Más adelante veremos qué hacer si el jugador introduce una palabra que no esté en el vocabulario suministrado a la máquina.

Ya que el programa trata todas las palabras de las direcciones como verbos, el mejor sitio para los verbos que indican lo que hacer con los objetos será la matriz R$, y el mejor sitio para los correspondientes números será la variable R.

En consecuencia tendrás que hacer algunas modificaciones, empezando por la línea 130. Tienes que cambiar

los límites del bucle FOR ... NEXT. La nueva versión de dicha línea es:

Teclea para C-64 y Vic-20

```
130 FOR K=1 TO 19:READ R$(K),
    R(K):NEXT
```

Seguidamente, añade las líneas 140 y 145:

```
140 DATA NADO,5,VACIO,6,LUZ,
    7,FIN,8,LISTA,9,MATO,10,
    DISPARO,10,AYUDO,11
145 DATA COJO,2,RECOJO,2,
    LLEVO,2,PONGO,3,DEJO,3,
    ABANDONO,3,TIRO,4
```

Cada verbo tiene su correspondiente número. Verbos con el mismo número tienen el mismo significado, por lo que al ordenador se refiere, y realizarán la misma operación. Por ejemplo, programando las cosas para que el ordenador acepte COGER, TOMAR y LLEVAR, el aventurero se ahorrará mucho de gasto de tiempo innecesario intentando descubrir cuál de estas palabras tiene que usar. Puedes añadir con facilidad tus propias palabras en las líneas de DATA cambiando el bucle FOR ... NEXT de la línea 130 y poniendo los nuevos DATA al final de la línea 145. Tienes que hacer más modificaciones en otras partes del programa, pero ya te diremos más adelante lo que tienes que hacer.

OTRAS RUTINAS

Después de que has completado la lista de verbos con la última rutina, el ordenador necesita algunas rutinas que le permitan atender instrucciones tales como hacer que el aventurero lleve determinados objetos.

La subrutina que comienza en la línea 3010 define V$, N$ e I, que es un número extraído de la matriz R. Con esta corta rutina, la máquina podrá comprender el significado de cada respuesta del aventurero, según el valor de I.

Teclea para C-64 y Vic-20

```
500 REM **ENCONTAR OPCION**
505 IF I=0 THEN GOTO 520
510 ON I GOTO 1010,1150,1240,
    1310,1410,1460,1500,1360,
    1080,1550,3110
520 PRINT:PRINT"NO CONOZCO ";
    V$:GOTO 370
```

Cada uno de los números que figuran después de la sentencia ON ... GOTO en la línea 510 es el principio de una subrutina. Cada valor de I corresponde a un verbo o grupo de verbos diferente. Por ejemplo, si I = 10, se seleccionará la rutina «matar», es el décimo número de la línea, por lo que la rutina comienza en la línea 1550.

Si la subrutina de comprobación de instrucciones, que comienza en la línea 3010, no encuentra coincidencias para la parte V$ de R$, se asigna a I el valor 0. En tal caso no tendrá efecto la sentencia ON ... GOTO de la línea 510 y se presentará el mensaje de la línea 520.

LA TOMA DE OBJETOS

Ya tienes la rutina para el I = 1, correspondiente al caso en que el aventurero introduce una palabra de dirección; dicha rutina se encuentra en las líneas 1010 a 1060.

Cuando I = 2, significa que el aventurero ha tecleado una palabra de «coger», como COGER, TOMAR o LLEVAR. La siguiente rutina permitirá al aventurero llevarse consigo cualquier objeto que haya en el lugar en que se encuentra. Sería algo así:

Teclea para C-64 y Vic-20

```
1140 REM **RUTINA "COJO"**
1150 FOR G=1 TO NB
1160 IF N$=LEFT$(OB$(G),LEN
     (N$)) THEN 1190
1170 NEXT
1180 PRINT " NO COMPRENDO ";
     N$:GOTO 330
1190 IF OB(G)=-1 THEN PRINT
     "YA LO TIENES":GOTO 330
1210 PRINT "[CTRL+5]OK[CTRL
     +8]":OB(G)=-1
1220 GOTO 330
```

En las líneas 1150 a 1170 se busca la matriz OB$ que contiene las descripciones cortas de objetos, para saber qué objeto es el designado por el aventurero. Si se encuentra el nombre del objeto, el programa salta a la línea 1190. Si el objeto no figura por ninguna parte de la aventura, la línea 1180 presenta el mensaje NO ENTIENDO, seguido del nombre de objeto tecleado por el aventurero.

Suponiendo que el objeto designado haya sido encontrado, hay que comprobar dos cosas. La línea 1190 examina el elemento de la matriz OB correspondiente a dicho objeto, para ver si ya está en poder del jugador. Si el jugador ya lo tiene (el correspondiente valor de la matriz es −1) se presentará el mensaje YA LO HAS COGIDO.

En la línea 1200 se comprueba si el objeto está presente, examinando nuevamente la matriz de lugares. Si no está presente, el programa dice: NO ESTA AQUI. Naturalmente, puedes cambiar estos mensajes por otros.

Si el objeto está en el mismo lugar que el aventurero y no ha sido cogido por éste, la línea 1210 dice OK y en el correspondiente elemento de la matriz de lugares se pone el valor −1.

ABANDONO DE OBJETOS

La rutina de «abandono» hace exactamente lo contrario que la anterior. Permite al aventurero dejar los objetos que no quiere llevar con él.

Teclea para C-64 y Vic-20

```
1230 REM **RUTINA "DEJA"**
1240 FOR G=1 TO NB
1250 IF N$=LEFT$(OB$(G),LEN
     (N$)) THEN 1270
1260 NEXT:PRINT"NO COMPRENDO
     "N$:GOTO 330
1270 IF OB(G)<>-1 THEN PRINT
     "NO LO TIENES":GOTO 330
1280 PRINT "OK":OB(G)=L
1290 GOTO 330
```

Esta rutina funciona de una forma muy parecida a la de «toma». Nuevamente se examina la matriz de descripciones cortas, esta vez se hace en las



P y R

¿Se puede utilizar un sintetizador de voz con una aventura?
Tu aventura podría resultar más interesante programando la máquina de modo que anuncie acústicamente los mensajes, direcciones y descripciones de objetos, en lugar de representarlos en la pantalla.

Consulta el manual de tu sintetizador para ver cómo se puede hacer que la máquina hable y sustituya a las instrucciones que contienen las sentencias PRINT.

líneas 1240 a 1260. Si el objeto designado por el aventurero figura en la matriz, la línea 270 comprueba si el aventurero lo lleva o no. Si no lo lleva, se presenta el mensaje NO LO LLEVAS.

Si el aventurero lleva el objeto, la línea 1280 envía el mensaje OK y se ajusta el correspondiente elemento en la matriz de situación de objetos OB. Ahora tiene el mismo valor que el lugar actual, es decir L, en vez de −1 que significaba que lo llevaba el aventurero.

INVENTARIO DEL BOTIN

Los aventureros desmemoriados estarán muy contentos de poder contar siempre que quieran con una lista de todos los objetos que llevan. Aquí tienes una rutina que hará eso:

Teclea para C-64 y Vic-20

```
1070 REM **LISTA**
1080 PRINT "[CTRL+4]TIENES:
     [CTRL+8] ";:IN=0
1090 FOR G=1 TO NB
1100 IF OB(G)=-1 THEN PRINT
     TAB(10)OB$(G):IN=IND+1
1110 NEXT
1120 IF IN=0 THEN PRINT
     "NADA"
1130 GOTO 330
```

La línea 1080 envía el mensaje LLEVAS, seguido de la lista de objetos. El bucle FOR...NEXT comprueba todos los elementos de la matriz de situación de objetos. Esta vez los elementos importantes son los que tienen el valor −1, significando que el correspondiente objeto es uno de los que lleva el aventurero. Si el valor de un elemento es −1, se presenta la descripción corta del objeto en cuestión, tomándola de la matriz. El contador de inventario IN se incrementa en 1.

Si el aventurero no lleva ningún objeto, IN se queda a cero y la línea 1120 presenta el mensaje NADA.

Las rutinas de «toma», «abandono» e «inventario» pueden utilizarse tal como están , ya que NB ha sido definido en una rutina anterior.

Almacena ahora el programa (SAVE), dejándolo dispuesto para recibir las rutinas finales que veremos la próxima vez. Se trata de las rutinas correspondientes al inspector de hacienda, el ladrillo, la lámpara, el encuentro de la joya, el final de la aventura y, finalmente, la instrucción que describe el objeto de la búsqueda.

Si ejecutas el programa ahora, verás que hay partes del mismo que funcionan mientras que en otras, no. La razón es que todavía hacen falta unas cuantas rutinas.

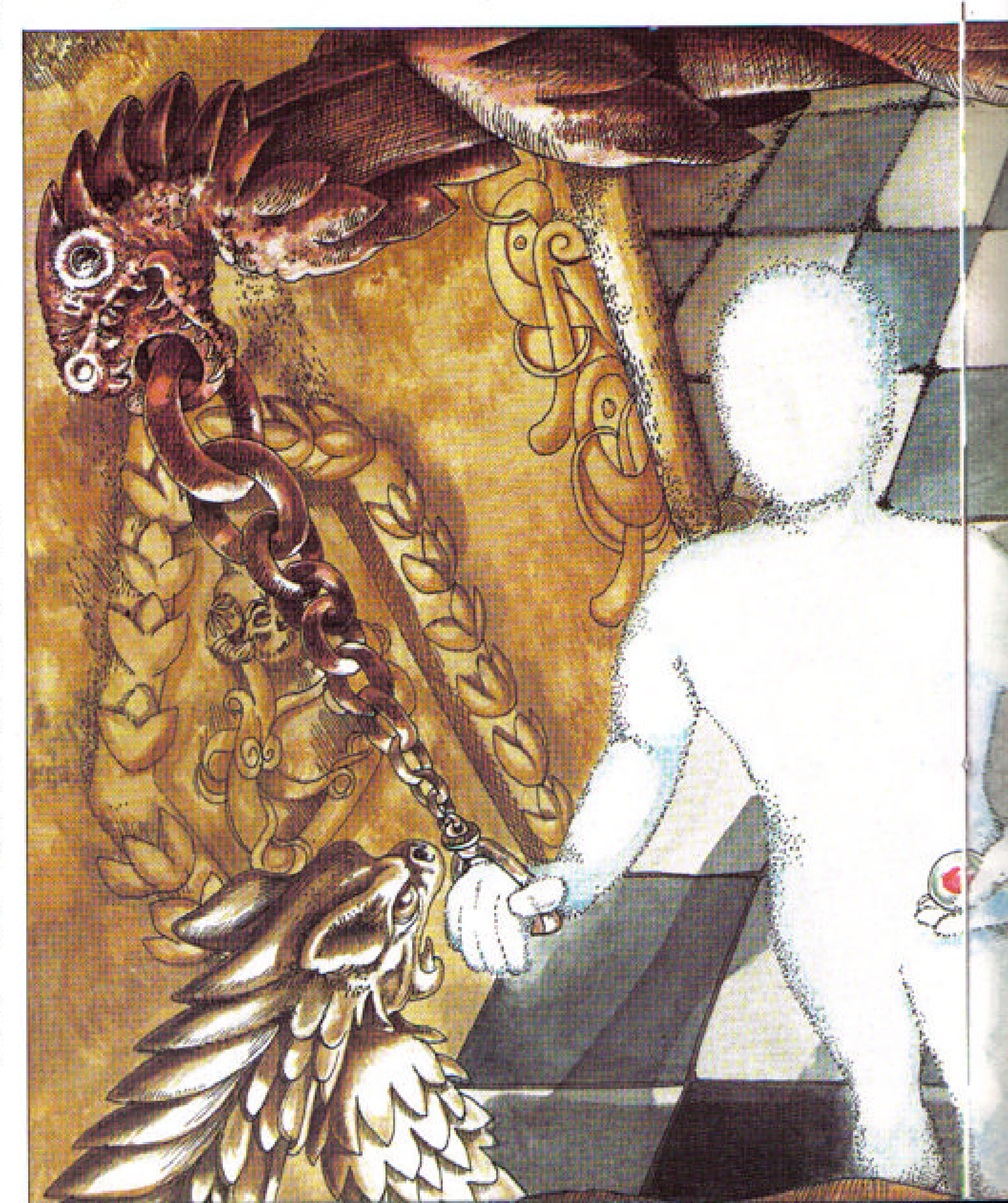
ELIMINANDO FALLOS

- Asegúrate de que los tres grupos DATA que contienen datos relacionados con los objetos se leen la matriz adecuada. Si intentas meter datos de cadenas de caracteres en una matriz numérica, recibirás un mensaje de error, o puede que te encuentres con una descripción corta cuando esperabas una larga.

- Ten mucho cuidado en respetar el orden de los DATA en correspondencia con el orden de lectura de las matrices. El orden correcto es: lugar, nombre o descripción corta, descripción larga.

- Después de introducir los objetos haz una pasada de prueba del programa, para asegurarte de que los objetos aparecen en el sitio correcto.

- Al comprobar los objetos, hazlo sobre la retícula, para asegurarte de que no se te ha olvidado nada.

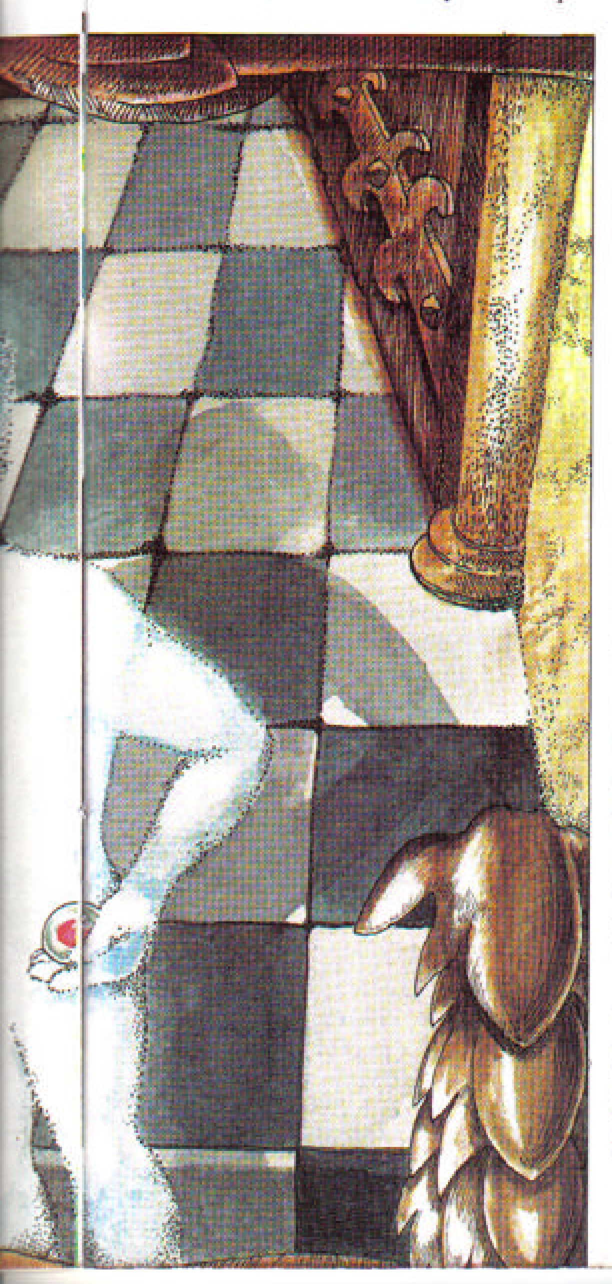


COMPLETANDO LA AVENTURA

- UNA RUTINA DE AYUDA
- EL INSPECTOR DE HACIENDA
- PROBLEMAS CON EL LADRILLO
- ENCENDIENDO LA LAMPARA
- INSTRUCCIONES

La aventura de INPUT está ya casi completa. Sólo quedan las rutinas que la convierten en una aventura única, las rutinas especiales que confieren a este juego su carácter.

Ya has completado casi todos los elementos que integran tu juego; es el momento de incorporar los últimos detalles del programa. Tienes que añadir aún algunas cosas, como los peligros, las advertencias, y tienes que prever un camino de salida para el caso de que la aventura se complete con éxito. Por último, tienes que poner las instrucciones para jugar.



Dado que muchas de estas rutinas están escritas de forma que contienen detalles específicos de una aventura particular, en los programas que siguen completaremos el juego y te mostraremos en términos generales todo lo que se relaciona con esta fase del programa. En otras aventuras no podrás utilizar estas rutinas sin modificarlas. En el próximo capítulo veremos la forma en que puedes adaptar estos principios para aplicarlos a tus ideas originales.

NECESITAS AYUDA

Si el juego de aventuras que has escrito es bueno, lo más probable es que el aventurero necesite alguna ayuda. Deberás pues prever unas cuantas sugerencias útiles.

Estas sugerencias tendrán la forma de mensajes que el ordenador enviará mediante sentencias PRINT en respuesta a las solicitudes de AYUDA hechas por el jugador. Lo que digan los mensajes y los puntos en que puedan estar disponibles, queda a tu entera discreción, ya que tú eres quien programa la aventura. Si quieres, puedes hacer que no haya un solo mensaje, o hacer que sean deliberadamente engañosos, o bien proporcionar ayuda únicamente en unas cuantas situaciones aisladas. El primer paso para decidir lo que incluir es volver a considerar tu plan original sobre las líneas maestras de la aventura.

En la aventura, hay varios puntos en los que convendría enviar al jugador un corto mensaje. Por ejemplo, podrías advertirle acerca de la habitación oscura, de forma que cuando esté en un lugar contiguo a dicha habitación, la respuesta ante su posible solicitud de ayuda podría ser un mensaje como: MIRA ANTES DE DAR EL SALTO, o incluso algo mucho más críptico.

Otro sitio donde se podría incluir un mensaje de aviso es la orilla del río, donde, en el caso de que se decida a nadar el aventurero corre el riesgo de ahogarse dependiendo de que lleve o no el ladrillo.

Naturalmente, puedes repasar una a una todas las situaciones en las que podría ser conveniente un poco de ayuda, pero supongamos que decides no ayudar demasiado y enviar un mensaje en un solo punto, el río. Tienes que hacer referencia a este número de lugar, el número 7, y a la variable que registra la presencia del ladrillo, OB(2):

Teclea para C-64 y Vic-20

```
3100 REM **RUTINA "AYUDA"**
3110 IF L<>7 OR OB(2)<>-1
     THEN PRINT "LO SIENTO,
     NO TE PUEDO AYUDAR AQUI
     !":GOTO 330
3120 PRINT "LOS LADRILLOS
     PESAN MUCHO Y HACEN QUE
     TEDUELA EL BRAZO":GOTO
     330
```

Si el aventurero no está en la orilla (L<>7), o no lleva consigo el ladrillo (OB(2)< > −1), la línea 3110 presentará el mensaje LO SIENTO, NO TE PUEDO AYUDAR AQUI. Si al llegar a la orilla del río, el aventurero lleva el ladrillo y pide ayuda, la línea 3120 imprimirá el siguienten mensaje de advertencia: LOS LADRILLOS PESAN MUCHO Y HACEN QUE TE DUELA EL BRAZO.

Caso de que quieras hacerlo, no hay nada que te impida incluir una lista completa de condiciones con sus correspondientes mensajes de aviso.

EL INSPECTOR DE HACIENDA

En la aventura interviene un sujeto que está por ahí merodeando, que es un inspector de hacienda el cual intenta recuperar parte de los impuestos impagados por el aventurero, confiscándole uno de los objetos que lleva. No se preocupa mucho de cuáles son esos objetos, por lo que en algunos casos incluso podría aceptar como pago un ladrillo.

Si el aventurero no tiene la suerte de llevar nada consigo en el momento en que se encuentra con el inspector de hacienda, será encerrado en una mazmorra en la que se pudrirá para siempre. Y aquí termina el juego.

El papel del inspector de hacienda es proporcionar al juego un elemento probabilístico, que resulte impredecible independientemente de cómo se encuentren las demás condiciones. Como en otros ejemplos de introducción de probabilidades en la programación, puedes hacerlo recurriendo a la función RND. Por lo demás, puedes tratarlo como otro objeto cualquiera; la única diferencia es que su situación no es fija, sino que se establece de forma aleatoria.

Aquí tienes las líneas suplementarias que añadir para que aparezca el inspector fiscal:

```
320 IF INT(RND(1)*15+1)=1
    AND TA=0 THEN OB(7)=L:
    TA=1
480 IF OB(7)=L AND I<>10
    THEN 1590
```

La línea 320 hace que el aventurero tenga una probabilidad sobre 15 de encontrarse con el inspector. Sólo se le permite aparecer una vez durante el juego, por lo que necesitas una variable, TA, para indicar si ha aparecido o no.

Si el número aleatorio es 1 o menor que 1/15, y el inspector todavía no ha aparecido, la línea 320 ajusta el valor que corresponde al inspector de hacienda en la matriz de situación de objetos. La presentación del mensaje del inspector se hace igual que si se tratase de un objeto, almacenándose en la descripción larga de la matriz.

La línea 480 tiene que ver con la supresión del inspector. Se limita a comprobar si has intentado matarle. Si no lo has intentado, el programa salta a la línea 1590.

LA RESPUESTA ANTE EL IMPUESTO

Cuando el inspector vuelve su fea cabeza, sólo hay una solución posible. El aventurero debe disparar contra él utilizando la pistola que se encontró en el río:

Teclea para C-64 y Vic-20

```
1540 REM **RUTINA "DISPARA
     "**
1550 IF OB(4)<>-1 THEN
     PRINT "CON QUE?":GOTO
     320
1560 IF OB(7)<>L THEN PRINT
     V$;" A QUIEN?":GOTO 320
1570 PRINT " MATASTE AL ";OB
     $(7):OB(7)=0:GOTO 330
```

Esta rutina se utiliza cuando el aventurero escribe las palabras MATAR o DISPARAR. Si no lleva la pistola (OB(4) <> −1), la línea 1550 presentará el mensaje CON QUE? Análogamente si el inspector de hacienda no está presente y el jugador intenta matarle, la línea 1560 le pregunta A QUIEN?

La línea 1570 le dice al aventurero HAS MATADO AL INSPECTOR DE IMPUESTOS, y ajusta la matriz de situaciones de los objetos de forma que dicho inspector ya no existe.

LA VENGANZA DEL INSPECTOR

El aventurero se encuentra con lo siguiente:

Teclea para C-64 y Vic-20

```
1580 REM **INSPECTOR
     FISCAL**
1590 IN=0:OB(7)=0
1600 FOR K=1 TO NB
1610 IF OB(K)=-1 THEN IN=
     IN+1
1620 NEXT
1630 IF IN<>0 THEN 1640
1635 PRINT "COMO NO LLEVAS
     NADA TE ENCIERRA "
1638 PRINT "EN UNA FRIA
     MAZMORRA":GOTO 1360
1640 K=INT (RND(1)*NB+1):IF
     OB(K)<>-1 THEN 1640
1650 PRINT "EL SE LLEVA EL "
     ;OB"(K):" LEJOS DE TI":
     OOB(K)=0: GOTO 330
```

Al inspector sólo se le permite aparecer una vez durante toda la aventura, por lo que la línea 1590 ajusta la matriz de situación de objetos en lo que afecta al programa. Esto no tiene efecto alguno en esta rutina, pero le ahorra aparecer también en el siguiente lugar. IN es un contador utilizado para comprobar si se llevan objetos.

En las líneas 1600 a 1620 se recorre toda la matriz de situaciones de objetos, para comprobar si cada uno de los objetos se lleva o no. Por cada objeto que se porte, se incrementa IN en uno.

Si el aventurero no lleva objeto alguno, el valor de IN permanece a cero y se presenta el siguiente mensaje: COMO NO LLEVAS NADA, TE ENCIERRA EN UNA FRIA MAZMORRA. El juego termina aquí, y se pregunta de nuevo al aventurero si quiere jugar otra vez saltando a la línea 1360.

Si por el contrario el aventurero lleva objetos consigo, en la línea 1640 se elige uno al azar. Si el número elegido corresponde a uno de los objetos transportados por el aventurero, dicho objeto es confiscado; si por el contrario dicho objeto no ha sido cogido, se selecciona otro número al azar, siguiendo así hasta que el número elegido corresponda a uno de los objetos transportados.

Después de que ha sido seleccionado un objeto, la línea 1650 informa al aventurero de que dicho objeto ha sido confiscado por el inspector. La matriz de situación de objetos queda modificada de forma que dicho objeto ya no existe.

LA ENCUESTA

De absoluta sorpresa ha sido vuestra reacción con respecto a la encuesta que publicamos en el anterior número de INPUT. Una auténtica avalancha de respuestas inunda las mesas de la redacción. El índice de receptividad ha sido extremadamente elevado y de ahí la sorpresa.

Según dicen los eruditos de la estadística aplicada, una muestra correctamente dimensionada proporciona información totalmente fiable sobre una población (antes lo llamaban universo). Aún no siendo expertos, creemos que la muestra proporcionada por vuestras cartas sera más que representativa. Cuando terminemos de tabularlas obtendremos una idea muy aproximada de las preferencias de la mayoría de los lectores.

De todas formas, leemos todos los cuestionarios, aunque sin gran detenimiento, y hemos logrado aislar algunas ideas que iremos llevando a la práctica poco a poco, para que la revista pueda identificarse cada vez más con sus lectores.

Otra sorpresa han sido las pocas críticas que nos destináis, aunque bien es cierto que hay algunas. Hay quienes decís que el nivel es alto, otros que bajo, pero lo verdaderamente difícil es escribir una revista de gran difusión que guste a todos. No obstante, invitamos a quienes podáis aportar algo de vuestros conocimientos a los demás. El equipo que elabora la publicación es gente joven a quienes gustan los ordenadores y hacer revistas. Las puertas están abiertas a todos vosotros. Participar tiene también sus ventajas. Sin ir más lejos, en este número aparece la lista de premiados por participar.

Otra conclusión provisional, ya intuida, es que la revista circula mucho de mano en mano. En gran cantidad de casos afirmáis compartir un ejemplar de la revista con otros usuarios.

Sólo queda agradeceros vuestra cooperación. En esta edición volvemos a publicar el cuestionario, antes de proceder al sorteo del ordenador.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Alberto Alcocer, 46 - 4.° B. 28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido ____

Segundo título elegido ____

Tercer título elegido ____

Programa que te gustaría conseguir ____

Qué ordenador tienes ____

Nombre ____

1.er Apellido ____

2.° Apellido ____

Fecha de nacimiento ____

Teléfono ____

Dirección ____

Localidad ____

Provincia ____

Programación

VARIABLES: TODOS LOS MISTERIOS EXPLICADOS

- ¿COMO SE USAN LAS VARIABLES?
- VARIABLES DE CONTROL
- VARIABLES NUMERICAS Y DE CADENAS, Y SUS USOS

Todas esas raras X e Y que aparecen en los programas son realmente muy sencillas de utilizar cuando se entiende lo que hacen. Ahora analizaremos la manera de utilizar las variables en tus propios programas.

Cuando quieres almacenar información en tu ordenador, tienes que decirle cómo puede identificarla. Si no lo haces así, el ordenador no podría encontrarla cuando vuelvas a requerirla.

Un tipo de información que se almacena temporalmente con mucha frecuencia son los números que incrementan o disminuyen su valor a medida que el programa avanza. Por ejemplo, podría tratarse de tu puntuación durante un juego o representar el número de espacios que un carácter se ha desplazado hacia la izquierda o la derecha en la pantalla o hacia arriba o abajo.

Muchas veces será simplemente un número cuyo valor cambia cada vez que el ordenador pasa por una sección del programa. Aquí tienes un sencillo ejemplo de esto:

```
10 LET X=0
20 LET X=X+1
30 PRINT X;" ";
40 FOR T=0 TO 10:NEXT T
50 GOTO 20
```

En la línea 50 se crea el bucle, haciendo que el programa se repita indefinidamente. Cada vez que llega a la línea 20, el número representado por X (que inicialmente había sido puesto a cero en la línea 10) se incrementa en una unidad.

La X de este programa es lo que se llama una **variable numérica,** ya que controla un número.

Una manera de representar la forma de trabajar de una variable numérica es imaginar una serie de cajas o compartimentos, como los que se usan para clasificar la correspondencia. Cada caja tiene un nombre determinado. En un ordenador sencillo, los nombres serían letras del alfabeto. Cuando quieras almacenar un número, no tienes más que asignar ese valor a una de las cajas. Por ejemplo:

LET C = 25 (haz C igual a 25).

Esta instrucción almacena el valor 25 en la variable C. No todos los ordenadores requieren que se utilice la palabra LET (la llamada sentencia de asignación), pero hasta que seas más experto en programación, es una buena práctica incluirla. Con ello te resultará algo más fácil la lectura de programas o evitar llevarte chascos si cambias de ordenador.

Una vez almacenado el valor numérico, se puede usar en otras sentencias y cálculos. Por ejemplo, podrías poner:

PRINT C*4

o bien:

PRINT C/5

Estas sentencias darían las respuestas 100 y 5 respectivamente. Pero si tecleas:

PRINT C

Verás que el valor de C ha permanecido sin alterar. Para cambiar el valor de C, por ejemplo, aumentándolo hasta 30, puedes hacerlo directamente tecleando

LET C=30

o tecleando

LET C=C+5

Naturalmente, esto es precisamente lo que hizo tu mini-programa original.

NOMBRES DE VARIABLES

Las combinaciones de caracteres permitidas como nombres de variables varían de un ordenador a otro. Más adelante veremos en un cuadro las principales reglas.

Pero una cosa que no te permitirá ningún ordenador es que el nombre de una variable comience con un número. Esto se debe a que sabe que

LET 7=14

no tiene sentido. Tampoco se te permite utilizar **palabras reservadas** del



BASIC, como GOTO, THEN y AND.

VARIABLES EN ACCION

La, mayoría de los programas que escribas utilizarán unas cuantas variables, con frecuencia interaccionando unas con otras. Aquí tienes, por ejemplo, un corto programa que simula la acción de un surtidor de gasolina:

Teclea para C-64 y Vic-20

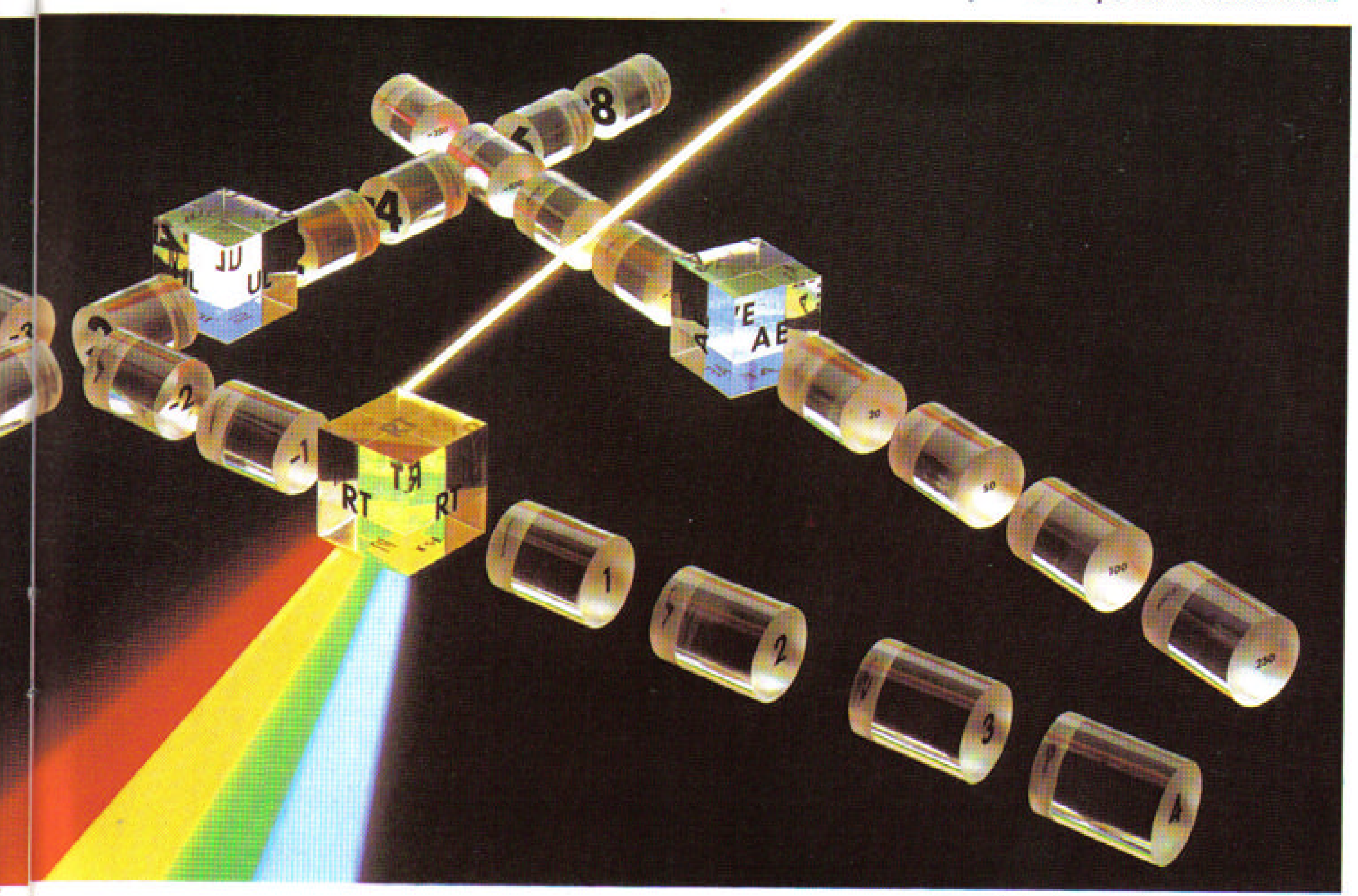
```
10 LET LITROS=0
20 LET PESETAS=0
30 PRINT"[SHIFT+CLR/HOME]
   [CRSR abajo]":POKE 650,
   128
40 PRINT"TECLEA: (90)NORMAL
   (96)SUPER (98)EXTRA"
50 INPUT A
60 IF A=90 THEN LET CANT=1/88
70 IF A=96 THEN LET CANT=1/91
80 IF A=98 THEN LET CANT=1/99
90 PRINT"[SHIFT+CLR/HOME]"
100 PRINT"[CLR/HOME][5*CRSR
    abajo][5*CRSR der.]";A;
    "OCTANOS"
110 PRINT"[CLR/HOME][7*CRSR
    abajo][6*CRSR der.]
    LITROS"
120 PRINT"[CLR/HOME][11*CRSR
    abajo][6*CRSR der.]
    IMPORTE"
130 GET K$
140 IF K$="F" THEN PRINT
    "[SHIFT+CLR/HOMRE]":
    GOTO 10
150 IF K$="N" THEN LET
    PESETAS=PESETAS+1:LET
    LITROS=LITROS+CANT
160 PRINT"[CLR/HOME][9*CRSR
    abajo][6*CRSR der.]";
    LITROS
170 PRINT"[clr/home][13*CRSR
    abajo][6*CRSR der.]PTS."
    PESETAS
200 GOTO 130
```

Conviene que experimentes con este programa hasta que entiendas completamente por qué tiene un conjunto tan completo de variables.

Lo que hacen algunas de ellas es evidente con sólo mirarlas. Por ejemplo, la variable A, sirve para ajustar el valor de una peseta al correspondiente valor de gasolina super, normal o extra.

A continuación, cuando pulsas N empiezas a salir la gasolina por la manguera y el reloj empieza a contar. Pesetas, que había sido puesto a cero en la línea 20, comienza a incrementarse de uno en uno (no se incluyen fracciones de peseta). Al mismo tiempo va aumentando la variable litros, que representa la cantidad de gasolina servida.

Cada vez que el programa llega a la línea 200, suponiendo que hayas mantenido pulsada la tecla N, se volverá a la línea 130 para mantener activo el reloj. Observa que no se vuelve a la lí-





nea 10, que pondría todo a cero otra vez.

Al ejecutar este programa observarás una cosa extraña: Algunas de las fracciones decimales que resultan son muy largas. Esto no quiere decir que tu máquina tenga un defecto. Es simplemente una de las limitaciones de los ordenadores en cuanto a su capacidad de pasar de la aritmética binaria a decimal.

VARIABLES DE CONTROL

Igual que se utilizan variables en sentencias LET, también se emplean en los bucles FOR... NEXT, como ya has visto sin duda muchas veces. Estas se llaman **variables de control.**

En algunos ordenadores, los caracteres que se pueden utilizar como varaibles de control son diferentes de los que se permiten para variables numéricas (ver cuadro más adelante).

CADENAS DE CARACTERES Y VARIABLES DE CADENA

Una cadena de caracteres de BASIC es algo parecido a la cesta de la compra. Puedes llenarla con los más variados artículos, que manejas como una única unidad, sin tener que mover cada uno de ellos por separado.

Las cadenas de caracteres pueden contener casi de todo: letras, números, espacios, incluso caracteres gráficos. Siempre han de estar contenidas entre las comillas de ambos extremos. Aquí tienes unos cuantos ejemplos:

«NOMBRE Y DIRECCION DEL CLIENTE»

«INTRODUZCA AHORA SU RESPUESTA»

«24 de enero»

«91-250 1000»

«28016»

El ordenador puede medir las cadenas de caracteres, es decir, contar el número de caracteres que la componen. También puede unir unas cadenas con otras (concatenar) o fraccionar una cadena en dos o más trozos cuando no se adapta a tu programa.

Pero el **contenido** de la cadena no le interesa para nada al propio ordenador. No puede sumarlas, excepto en el sentido ya indicado de unir una junto a otra, como tampoco puede multiplicarlas o dividirlas.

Para probar esto —y también para desconcertar a tus amigos no poseedores de un ordenador— prueba este corto programa:

Teclea para C-64 y Vic-20

```
10 PRINT"2+2= ";
20 FOR N=1 TO 200:NEXT N
30 PRINT1+2*2
```

Como puedes ver, el contenido de la cadena se imprime exactamente como lo has escrito. Sólo se hace un cálculo en la línea 30, imprimiéndose su resultado. (La línea 20 sirve para hacer que el ordenador «se rasque la cabeza» antes de dar una respuesta).

Si quieres usar la cadena solo una vez dentro de un programa, esto es todo lo que tienes que saber. Pero si quieres utilizar una cadena de caracteres más de una vez, puedes ahorrarte tiempo de mecanografiado y espacio de memoria, asignándole una etiqueta a tu cadena. Esta etiqueta se llama una **variable de cadena.** Su longitud varía de unos ordenadores a otros (ver cuadro), pero debe ir seguida siempre del signo $.

Aquí tienes, por ejemplo, el esquema básico de un sistema de pedido automático:

Teclea para C-64 y Vic-20

```
10 LET A$="POR FAVOR,
   ESCRIBE CUANTAS "
20 LET B$=" QUIERES"
30 PRINT A$;"HAMBURGUESAS";
   B$:INPUT B
40 PRINT A$;"PATATAS FRITAS"
   ;B$:INPUT P
50 PRINT A$;"BEBIDAS";B$:
   INPUT C
60 PRINT"GRACIAS, ME DEBES "
   ;B*180+P*85+C*110;
   "PESETAS"
```

No se trata de un sistema de pedidos especialmente bueno; sería mejor uno de tipo visual. Pero sirve para ilustrar cómo las cadenas pueden acortar un programa.

Por esta razón los sistemas de proceso de textos y de datos utilizan extensamente variables de cadena del tipo «número de cuenta del cliente», «precio por millar», «más el 15% de IVA» y otras cosas por el estilo que se presentan con mucha frecuencia.

Pero también puedes usar las cadenas en la programación de juegos. Supón que tienes una larga línea de símbolos gráficos, por ejemplo, la pared de una mazmorra en un juego de aventuras. No tienes más que teclearlos una vez, «etiquetarlos» y ya puedes utilizarlos tantas veces como quieras.

Además puedes desplazar la cadena por la pantalla, como te muestra el siguiente programa:

Teclea para Commodore-64

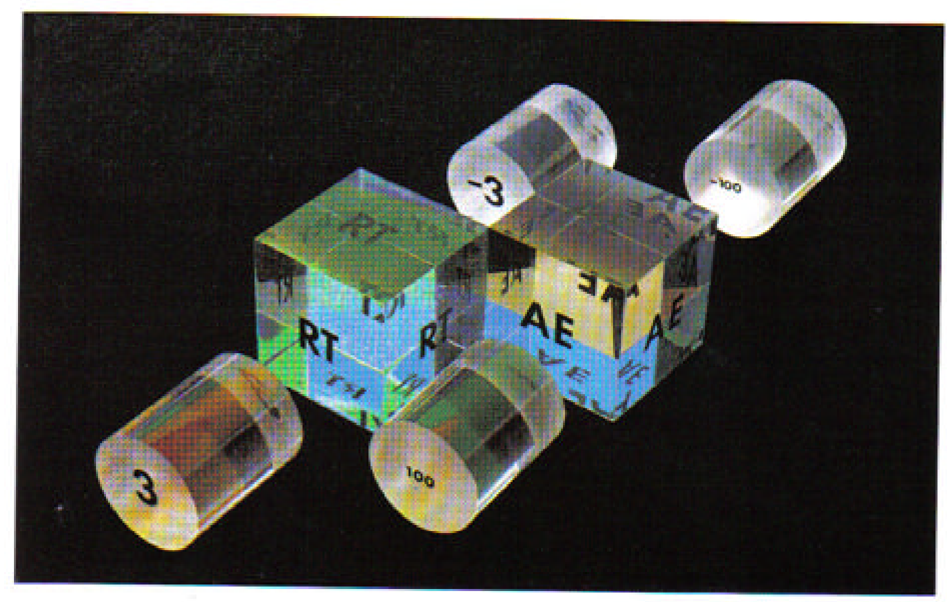
```
10 PRINT "[SHIFT+CLR/HOME]"
15 POKE 54277,33:POKE 54278,
   255:POKE 54273+23,15:POKE
   54276,33
20 A$="[5*ESPACIO]FELIZ
   NAVIDAD A TODOS[36*
   ESPACIO]":B$=A$
40 A$=RIGHT$(A$,61)+LEFT$
   (A$,1)
45 PRINT"[CTRL+6][CLR/HOME]
   [8*CRSR abajo]"LEFT$
   (A$,40)
50 B$=RIGHT$(B$,1)+LEFT$
   (A$,61)
55 PRINT"[ctrl+8][6*CRSR
   abajo]"LEFT$(B$,40)
60 S=S+5:IF S>255 THEN S=0
70 POKE 54273,S
80 GOTO 40
```

Teclea para Vic-20

```
10 PRINT"[CLR/HOME]"
15 POKE 36878,15:S=127
20 A$="[6*ESPACIO]FELICES
   PASCUAS[23*ESPACIO]":
   B$=A$
40 A$=RIGHT$(A$,61)+LEFT$
   (A$,1)
45 PRINT"[CTRL+5][CLR/HOME]
   [8*CRSRabajo]"LEFT$(A$,22)
50 B$=RIGHT$(B$,1)+LEFT$
   (B$,61)
55 PRINT"[CTRL+8][6*CRSR
   abajo]"LEFT$(B$,22)
60 LET S=S+5:IF S>255 THEN
   S=128
70 POKE 36876,S
80 GOTO 40
```

VARIABLES: Lo que puedes hacer y lo que no.

TIPO DE VARIABLE	COMMODORE 64, VIC-20
Variable numérica	Longitud máxima 255 caracteres, pero sólo serán reconocidos los dos primeros. Sólo se permiten mayúsculas, pero no pueden empezar por palabras clave, p. ej., TO por TOTAL. No se permiten signos de puntuación. Se permiten espacios
Variable de control en bucles FOR	Una sola letra, o letra seguida de un carácter, p. ej.: A o A3
Variable de cadena	Como las variables numéricas pero añadiendo $, p. ej.: AD$



Como muestra la línea 20, tienes una larga cadena, todos los espacios a partir de la línea 10, más el mensaje, más todos los espacios de nuevo, agrupados como un todo. En un próximo artículo veremos cómo se hace esto y cómo se «trocea» la cadena para caber en la pantalla. Pero entretanto desearás cambiar el mensaje, asegurarte de que el nuevo tiene la misma longitud que el original. De no ser así, tendrías que modificar los números que intervienen en las líneas 40 a 55.

Un uso aún más común de las variables de cadena es el que se produce cuando el ordenador espera información que cambia constantemente. Por ejemplo:

Teclea para C-64 y Vic-20

```
10 PRINT "COMO TE LLAMAS"
20 INPUT N$
30 PRINT "ADIOS, ";N$
```

La línea 10 de este programa imprime la pantalla la cadena de caracteres ¿COMO TE LLAMAS? Seguidamente el ordenador espera a que introduzcas tu nombre, constituido por una serie de letras que forman otra cadena de caracteres, a la que hemos llamado N$.

En la línea 30 se imprime la cadena «HOLA» y llama a la cadena N$ que almacenó en la línea anterior. En la pantalla aparecerá.

HOLA, JUAN (o LUIS, o JAVI)

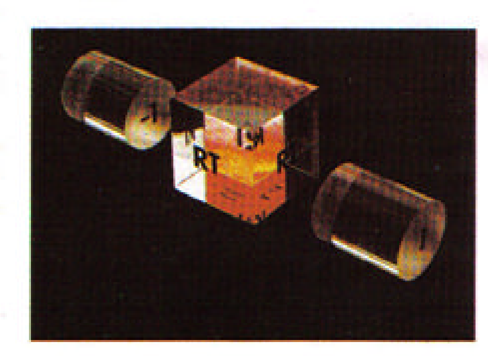
o cualquier otro nombre que hayas puesto.

Observa que a las cadenas ¿COMO TE LLAMAS? y HOLA no se les ha asignado un nombre como A$ o COMO$ u HOLA$, ya que sólo aparecen una vez y el ordenador no necesita recordar lo que es HOLA. Cada vez que llega a la línea 20 se limita a leerla y a imprimir lo que contiene.

Sin embargo, el nombre introducido no necesariamente tiene que llevar una etiqueta. Puede ser diferente cada vez que se ejecuta el programa. También en este caso se almacena en la línea 20, pudiendo ser llamado por la línea 30.

Por último, aquí tienes una versión más corta de este último programa, ligeramente «ensanchada» por razones de claridad. La forma corta es:

```
10 INPUT "COMO TE LLAMAS ";
   N$
```

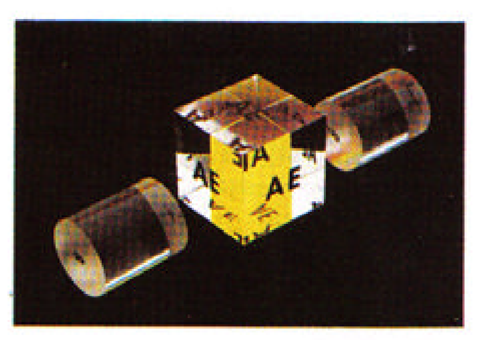


```
20 PRINT "ADIOS, ";N$
```

CADENAS NULAS

También hay otro uso para las variables de cadena, que puede que hayas encontrado en la programación de juegos. Varía ligeramente de unas máquinas a otras, pero su apariencia es más o menos la siguiente:

```
20 IF A$=""THEN GO TO 10
```



Las comillas sin nada en medio constituyen la **cadena nula.** Significan «si la entrada es igual a nada»; en otras palabras. «si no se ha pulsado tecla alguna», en cuyo caso vuelve a la línea 10 y espera hasta que se pulse alguna.

Esto evita que el ordenador pase por el programa tan rápidamente que simplemente el jugador no tenga ni siquiera tiempo de pulsar una de las teclas.

Una cosa muy distinta son las dos comillas con un espacio entre ellas. Si modificas la línea de la forma siguiente:

```
20 IF A$="[ESPACIO]"THEN
   GOTO 10
```

El programa sólo volverá a la línea 10 si el jugador introduce un espacio, es decir si ha pulsado la tecla espaciadora.

GANADORES DE LOS MEJORES DE INPUT COMMODORE

En el sorteo correspondiente al número 5 realizado entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
José Martín López	Barcelona	Karateka
Joan Ferré Navarro	Barcelona	Archon
Luis Miguel Gorgues	Lérida	Karateka
Rafael Berenguer Vidal	Onil	Skyfox
José Noguera Camprubí	Berga	Summer Games
Joaquín Embil López	Mieres	Skyfox
Víctor Rodríguez Calzada	Palencia	Fight night
Borja Martínez de la Rosa	Barcelona	Commando
Jesús M.ª González García	Vitoria	Skyfox
Jordi García Pujadas	Manresa	Skyfox

¡ATRAPADOS EN EL REINO DE LO IMPOSIBLE!

REALM OF IMPOSSIBILITY

C-64 - Amstrad

MIKE EDWARDS'
REALM OF IMPOSSIBILITY
REINO DE LO IMPOSIBLE
Sistema único de cooperación entre dos jugadores
DRO SOFT
ELECTRONIC ARTS

Por lo visto, las arañas sirven a AHADDON. La pregunta es: ¿AMAN [illegible] las arañas? Es desagradable tener a muchas [illegible] esperando para bailar alrededor de tu pequeño cadáver. ¡Simplemente insufrible!

Acorralado en la habitación caverna de las CRYPTAS ESTIGIAS. Si tan sólo tuvieras un hechizo de congelación o de protección, o incluso un hechizo de confusión... ¡Cualquier cosa que te mantuviera vivo!

La ayuda de un amigo y mucha suerte es todo lo que tienes para salir con vida. Los Zombies, Orbs, Arañas y Serpientes prefieren que te unas a ellos en su fantasmal vagar por toda la eternidad.

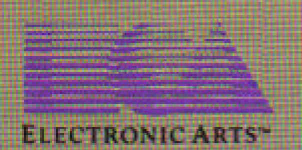
Sistema único de cooperación entre dos jugadores, unidos para luchar contra los horrores del Reino Imposible. 13 cavernas distintas, 129 habitaciones diferentes. ¡ACCION increíble! 4 niveles de dificultad. Posibilidades de juego para un solo jugador.

Olvida los otros juegos de pantallas en solitario. Dos jugadores lo hacen DIVERTIDISIMO en vez de divertido.

UNA PESADILLA EN LA PANTALLA

Editado por DRO SOFT. Fundadores, 3. 28028 Madrid. Telf. 255 45 00/09





Electronic Arts: Somos una asociación de Artistas de la electrónica que comparten una meta común: Queremos explotar al máximo el uso personal del ordenador. Es algo difícil de llevar a cabo. Pero con la suficiente imaginación y entusiasmo creemos que hay verdaderas posibilidades de éxito. Nuestros productos, como estos juegos, son una prueba evidente de nuestro esfuerzo.

FORMAS DE LA NATURALEZA

- MODELOS DE SIMETRIA
- PROGRAMA DE COPOS DE NIEVE
- LA MONTAÑA COMPUTERIZADA
- GENERACION DE FORMAS

Desde las formas idealizadas de las fractales generadas matemáticamente a formas capaces de modelar el mundo natural con verdadero realismo.

En el primer artículo sobre fractales, vimos cómo se pueden usar unos cuantos sencillos programas recursivos para generar formas fascinantes, mediante la aplicación repetida de una sencilla operación. Estas formas generadas matemáticamente presentan una imagen de simetría y orden, pero aunque tienen muchos puntos en común con las formas irregulares que se encuentran en la Naturaleza parecen poco más que meras curiosidades.

Aunque ese tipo de fractales se acerca al modelado del mundo natural mucho más de lo que es posible usando las formas perfectas de la ciencia tradicional, siguen estando aún bastante alejadas del realismo. En este artículo veremos la manera en que las fractales pueden modelar las formas que se encuentran en la Naturaleza.

El primer programa presenta una de las formas más simétricas de la Naturaleza, el hexagonal copo de nieve.

Teclea para Commodore 64

```
1 REM**UTILIZAR EL CARTUCHO
  SIMON'S BASIC
10 HIRES 0,1:MULTI 0,1,7:
   COLOUR 5,5
20 AN=2*ATN(1)/3:S2=2/SQR(3)
30 XC=80:YC=99:S=99:C=4
50 GOSUB 1000:FOR D=1 TO
   5000:NEXT D:COLOUR 0,0
60 MULTI RND(1)*16,RND(1)*16
   ,RND(1)*16:GOTO 60
1000 S=S/3:IF S<1 THEN S=S*3
     :RETURN
1010 CL=CL+1:IF CL>3 THEN
     CL=1
1020 XX=INT(XC+S2*S*SIN(-AN)
     ):YY=INT(YC-S2*S*COS
     (-AN))
1025 FOR K=0 TO 8*ATN(1)-AN
     STEP 2*AN
1030 LINE XX,YY,XC+2*S*SIN
     (K),YC-2*S*COS(K),CL
1040 LINE XC+2*S*SIN(K),YC-2
     *S*COS(K),XC+S2*S*SIN
     (K+AN),YC-S2*S*COS(K+AN
     ),CL
1045 XX=XC+S2*S*SIN(K+AN):
     YY=YC-S2*S*COS(K+AN)
1050 NEXT K
1060 C=C-1:GOSUB 1000
1070 YC=YC-1.36*S:GOSUB 1000
1080 YC=YC+.68*S:XC=XC+1.19*
     S:GOSUB 1000
1090 YC=YC+1.36*S:GOSUB 1000
1100 YC=YC+.68:XC=XC-1.18*S:
     GOSUB 1000
1110 YC=YC-.68*S:XC=XC-1.19*
     S:GOSUB 1000
1130 YC=YC+.68*S:XC=XC+1.19*
     S:S=S*3:C=C+1:RETURN
```

El listado de programa para el **Vic 20** es igual que el del **Commodore 64**, excepto las siguientes líneas:

Teclea para Vic-20

```
1 REM**UTILIZAR EL CARTUCHO
  DE AMPLIACION DEL BASIC
10 GRAPHIC 2:COLOR 0,0,1,1
20 AN=ATN(1)/3:S2=2/SQR(3)
30 XC=512:YC=512:S=700
50 GOSUB 1000
60 MULTI RND(1)*16,RND(1)*16
   ,RND(1)*16:GOTO 60
1000 S=S/3:IF S<10 THENS=S*3
     :RETURN
1010 CL=CL+1:IF CL>7 THEN
     CL=1
1020 POINT 0,INT(XC+S2*S*SIN
     (-AN)),INT(YC-S2*S*COS
     (-AN)):REGION(CL)
1025 FOR K=0 TO 8*ATN(1)-AN
     STEP 2*AN
1030 DRAW 1 TO XC+2*S*SIN
     (K),YC-2*S*COS(K)
1040 DRAW 1 TO XC+2*S*SIN(K)
     ,YC-2*S*COS(K),XC+S2*S*
     SIN(K+AN)
1045 XX=XC+S2*S*SIN(K+AN):
     YY=YC-S2*S*COS(K+AN)
1050 NEXT K
1060 C=C-1:GOSUB 1000
1070 YC=YC-1.36*S:GOSUB 1000
1080 YC=YC+.68*S:XC=XC+1.19*
     S:GOSUB 1000
1090 YC=YC+1.36*S:GOSUB 1000
1100 YC=YC+.68:XC=XC-1.18*S:
     GOSUB 1000
1110 YC=YC-.68*S:XC=XC-1.19*
     S:GOSUB 1000
1130 YC=YC+.68*S:XC=XC+1.19*
     S:S=S*3:C=C+1:RETURN
```

Este programa se basa en la curva del copo de nieve, dibujada originalmente por **von Koch**. Puede considerarse como un copo de nieve infinitamente arrugado, o bien como una isla cuya línea de costa tiene longitud infinita. El programa especifica en la línea 20 un triángulo equilátero, cuyo factor de escala se ha fijado en S2. La línea 30 establece coordenadas X e Y para el centro, un factor de escala para el tamaño de cada triángulo y una variable para cambiar el color. En la línea 50 se llama a una subrutina recursiva que dibuja una figura con forma de estrella de seis puntas.

SIMETRIA Y CAOS

A pesar de la irregularidad del contorno, se mantiene gran parte de la regularidad de la estrella. Se necesita la simetría para modelar formas como la del copo de nieve, que combinan el caos y el orden; dichas formas son comunes en la Naturaleza. Pero muchas estructuras naturales que pueden entenderse mejor como fractales carecen totalmente de simetría. Son ejemplos de esto las curvas del río Mississippi, la superficie de las escamas de jabón,



los agujeros de un queso suizo, los cráteres de la Luna, las venas y arterias del cuerpo humano y las formas de las montañas. Lo que distingue a estos ejemplos de las formas simétricas generadas matemáticamente es que también contienen un grado de aleatoriedad. Pero también es posible generarlas, utilizando el propio generador de números aleatorios del ordenador. Aquí tienes por ejemplo, un programa para modelar una montaña:

Teclea para C-64

```
1 REM UTILIZAR CARTUCHO
  SIMON'S BASIC
10 HIRES 0,1:MULTI 5,13,0:
   COLOUR 6,14:X=1
15 BLOCK 0,100,159,199,3:FOR
   Z=1 TO 80:PLOT RND(1)*160
   ,RND(1)*100,2:NEXT Z
16 FOR Z=1 TO 20:LINE 0,100+
   XX,159,100+XX,9:X=X+.4:XX
   =XX+X:NEXT Z
20 DIM C(200,1,1):F=0:G=1:C
   (0,0,1)=150:C(0,0,0)=10
30 S=50:L=140/S:H=L/SQR(2)
40 FOR K=1 TO S:C(K,0,0)=C(0
   ,0,0)+L*K-3+RND(1)*6
45 C(K,0,1)=C(K-1,0,1)-3+RND
   (1)*6:NEXT K
50 FOR J=1 TO S:FOR K=0 TO
   S-J
60 C(K,G,0)=3-RND(1)*6+(C(K,
   F,0)+C(K+1,F,0))/2
70 C(K,G,1)=3-RND(1)*6-H+(C(
   K,F,1)+C(K+1,F,1))/2
```

DATA BECKER

FERRE MORET, S.A.
C/. TUSET, 8-10, ENTL. 2.
TEL. 218 02 93 - 08006 BARCELONA

Estos son nuestros BESTSELLERS

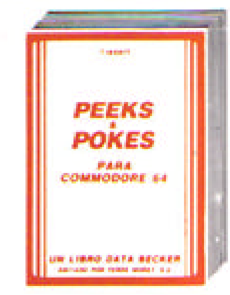




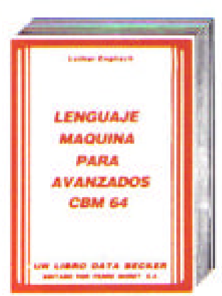




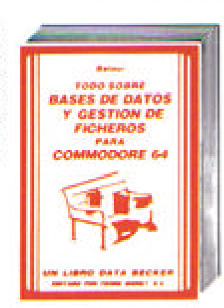




OTROS TITULOS



¿Ud. ha logrado iniciarse en código máquina? Entonces el «nuevo English» le enseñará cómo convertirse en un profesional. Naturalmente con muchos programas ejemplo, rutinas completas en código máquina e importantes consejos y trucos para la programación en lenguaje máquina y para el trabajo con el sistema operativo.
Lenguaje máquina para avanzados CBM 64. 1984, 206 pág. P.V.P. 2.200 ptas.



Este libro ofrece una amplia práctica introducción en el importante tema de la gestión de ficheros y bancos de datos, especialmente para los usuarios del Commodore 64. Con muchas interesantes rutinas y una confortable gestión de ficheros.
Todo sobre bases de datos y gestión de ficheros para Commodore-64. 221 págs. P.V.P. 2.200,- ptas.



Gráficos para el Commodore 64 es un libro para todos los que quieren hacer algo creativo con su ordenador. El contenido abarca desde los fundamentos de la programación de gráficos hasta el diseño asistido por ordenador (CAD).
Gráficos para el Commodore 64. 295 pags. P.V.P. 2.200,- ptas.



Para los usuarios que posean un VIC-20, C-64 o PC-128 este libro contiene gran cantidad de consejos, trucos, listados de programas, así como información sobre Hardware, tanto si usted dispone de una impresora de margarita o de matriz, como si tiene un Plotter VC-1520, el GRAN LIBRO DE IMPRESORAS constituye una inestimable fuente de información.
Todo sobre impresoras. 361 págs. P.V.P. 2.800,- ptas.



Este libro presenta una detallada e interesante introducción a la teoría, conceptos básicos y posibilidades de uso de la inteligencia artificial (IA). Desde un resumen histórico sobre las máquinas «pensantes» y «vivientes» hasta programas de aplicación para el Commodore 64.
Inteligencia artificial. 395 págs. 2.800,- ptas.



64, Consejos y Trucos vol. 2 contiene una gran profusión de programas, estímulos y muchas rutinas útiles. Un libro que constituye una ayuda imprescindible para todo aquél que quiera escribir programas propios con el COMMODORE.
Consejos y Trucos, Commodore 64. Vol. 2. 259 págs. 2.200,- ptas.



En el libro de los robots se muestran las asombrosas posibilidades que ofrece el CBM 64, para el control y la programación, presentadas con numerosas ilustraciones e intuitivos ejemplos. El punto principal: Cómo puede construirse uno mismo un robot sin grandes gastos. Además, un resumen del desarrollo histórico del robot y una amplia introducción a los fundamentos cibernéticos. Gobierno del motor, el modelo de simulación, interruptor de pantalla, el Port-Usuario cómodo del modelo de simulación, Sensor de infrarrojos, concepto básico de un robot, realimentación unidad cibernética, Brazo prensor, Oír y ver.
Robótica para su Commodore 64. 340 págs. P.V.P. 2.800 ptas.



Saberse apañar uno mismo, ahorra tiempo, molestias y dinero, precisamente problemas como el ajuste del floppy o reparaciones de la platina se pueden arreglar a menudo con medios sencillos. Instrucciones para eliminar la mayoría de perturbaciones, listas de piezas de recambio y una introducción a la mecánica y a la electrónica de la unidad de disco, hay también indicaciones exactas sobre herramientas y material de trabajo. Este libro hay que considerarlo en todos sus aspectos como efectivo y barato.
Mantenimiento y reparación del Floppy 1541. 325 págs. P.V.P. 2.800,- ptas.



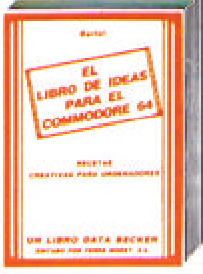
Este es el libro que buscaba: un diccionario general de micros que contiene toda la terminología informática de la A a la Z y un diccionario técnico con traducciones de los términos ingleses de más importancia - los DICCIONARIOS DATA BECKER prácticamente son tres libros en uno. La increíble cantidad de información que contienen, no sólo los convierte en enciclopedias altamente competente, sino también en herramientas indispensables para el trabajo. El DICCIONARIO DATA BECKER se edita en versión especial para APPLE II, COMMODORE 64 e IBM PC.
El diccionario para su Commodore 64. 350 pág. P.V.P. 2.800,- ptas.



Ofrece un campo fascinante y amplio de problemáticas científicas. Para esto el libro contiene muchos listados interesantes: Análisis de Fornier y síntesis, análisis de redes, exactitud de cálculo, formateado de números, cálculo del valor PH, sistemas de ecuaciones diferenciales, modelo ladrón presa, cálculo de probabilidad, medición de tiempo, integración, etc.
64 en el campo de la Técnica y la Ciencia. 361 págs. P.V.P. 2.800,- ptas.



Este libro, contiene muchos interesantes programas de aprendizaje para solucionar problemas, descritos detalladamente y de manera fácilmente comprensible. Temas: progresiones geométricas, palanca mecánica, crecimiento exponencial, verbos irregulares, ecuaciones de segundo grado, movimientos de péndulo, formación de moléculas, aprendizaje de vocablos, cálculo de interés y su capitalización.
Manual escolar para su Commodore 64. 389 págs. P.V.P. 2.800,- ptas.



Casi todo lo que se puede hacer con el Commodore 64, está descrito detalladamente en este libro. Su lectura no es tan sólo tan apasionante como la de una novela, sino que contiene, además de listados de útiles programas, sobre todo muchas, muchas aplicaciones realizables en el C64. En parte hay listados de programas listos para ser tecleados, siempre que ha sido posible condensar «recetas» en una o dos páginas. Si hasta el momento no sabía que hacer con su Commodore 64, ¡después de leer este libro lo sabrá seguro!
El libro de ideas del Commodore 64. 1984, más de 200 páginas, P.V.P. 1.600,- ptas.



Este libro ofrece al programador interesado una introducción fácilmente comprensible para los tan extendidos Assembler PROFI-ASS, SM MAE y T.E-X.ASS, con consejos y trucos de gran utilidad, indicaciones y programas adicionales. Al mismo tiempo sirve de manual orientado a la práctica, con aclaraciones de conceptos importantes e instrucciones.

El Ensamblador. 250 páginas. 2.200,- ptas.



El libro de Primicias del Commodore 128 no ofrece solamente un resumen completo de todas las características y rendimientos del sucesor del C-64 y con ello una importante ayuda para su adquisición. Muestra, además, todas las posibilidades del nuevo equipo en función de sus tres modos de operación.

Todo sobre el nuevo Commodore 128. 250 págs. P.V.P. 2.200,- ptas.



El libro Commodore 128-Consejos y Trucos es un filón para cualquier poseedor del C-128 que desee sacar más partido a su ordenador. Este libro no sólo contiene gran cantidad de programas-ejemplo, sino que además explica de un modo sencillo y fácil la configuración del ordenador y de su programación.

Commodore 128-Consejos y Trucos. 327 págs. 2.800,- ptas.

OTROS TITULOS



El primer libro recomendado para escuelas de enseñanza de informática y para aquellas personas que quieren aprender la programación. Cubre las especificaciones del Ministerio de Educación y Ciencia para Estudios de Informática. Es el primer libro que introduce a la lógica del ordenador. Es un elemento de base que sirve como introducción para la programación en cualquier otro lenguaje. No se requieren conocimientos de programación ni siquiera de informática. Abarca desde los métodos de programación clásicos a los más modernos.

Metodología de la Programación. 250 págs. P.V.P. 2.200,- ptas.



La técnica y programación del Procesador Z80 son los temas de este libro. Es un libro de estudio y de consulta imprescindible para todos aquellos que poseen un Commodore 128, CPC, MSX u otros ordenadores que trabajan con el Procesador Z80 y desean programar en lenguaje máquina.

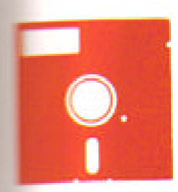
El Procesador Z80. 560 pág. P.V.P. 3.800,- ptas.



El tema de este libro es la técnica y programación de los procesadores de la familia 68000. Es una obra de consulta indispensable, un manual para todo programador que quiera utilizar las ventajas del 68000.

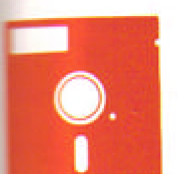
Técnica y programación para el procesador 68000. 516 págs. P.V.P 3.800,- ptas.

SOFTWARE COMMODORE



PLATINE 64 - LA LLAVE PARA EL MERCADO ELECTRONICO

Hasta ahora los sistemas de desarrollo de placas de circuito impreso (platinas) sólo eran accesibles con mucho dinero y a base de computadoras caras. De allí que había muchos, que necesitaban un sistema de este tipo, pero pocos que podían costeárselo. Esto ha llegado a su fin. Compare los datos, PLATINE 64 a este precio y en esta categoría de ordenadores, es una auténtica sensación a nivel mundial. Puede solicitarse folleto de información.*

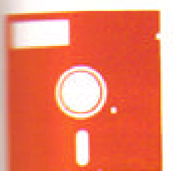


PROFIMAT 5.850 ptas.

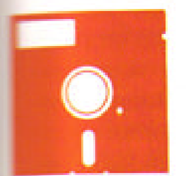
Quien quiere acceder a la vida interior del computador, necesita disponer de herramientas especiales. Aquí es donde PROFIMAT entra en acción, ofreciendo un confortable monitor de código máquina así como un macroensamblador.



KALKUMAT 11.800 ptas.

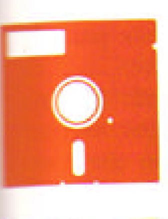


El KALKUMAT es una hoja de càlculo electrónica, que puede aplicarse para efectuar cualquier tipo de cálculos, estadísticas, proyectos de planings y simulaciones.
El paquete del KALKUMAT contiene el programa KALKUGRAF, que permite representar los cálculos mediante gráficas de sectores y de barras de dos y tres dimensiones, o mediante diagramas mínimos y máximos o representaciones lineales. La pantalla está estructurada en cuatro zonas: entrada, contenido, hoja de trabajo y menú, que le indican continuamente la situación actual del programa.



ADA — curso de entrenamiento 11.700 ptas.

Este lenguaje de programación del futuro, el cual al igual que COBOL fue encargado por el pentágono, ahora puede ser aprendido por el usuario del C-64, mediante el curso de entrenamiento de DATA BECKER. El curso de entrenamiento de ADA además contiene un compilador que ofrece un exhaustivo SUBSET, y los elementos principales de este lenguaje.



ELECTROMAT 4.100 ptas.

Con ELECTROMAT, la creación de esquemas de circuitos electrónicos se convierte en una diversión todos los esquemas pueden editarse directamente en pantalla.



TEXTOMAT 5.850 ptas

¡El procesador de textos más vendido en Alemania, ahora también disponible para Armstrad CPC 464!

BOLETIN DE PEDIDO

FERRE - MORET S.A. Tuset n.° 8, entlo. 2.° BARCELONA 08006 Tel. 218 02 93

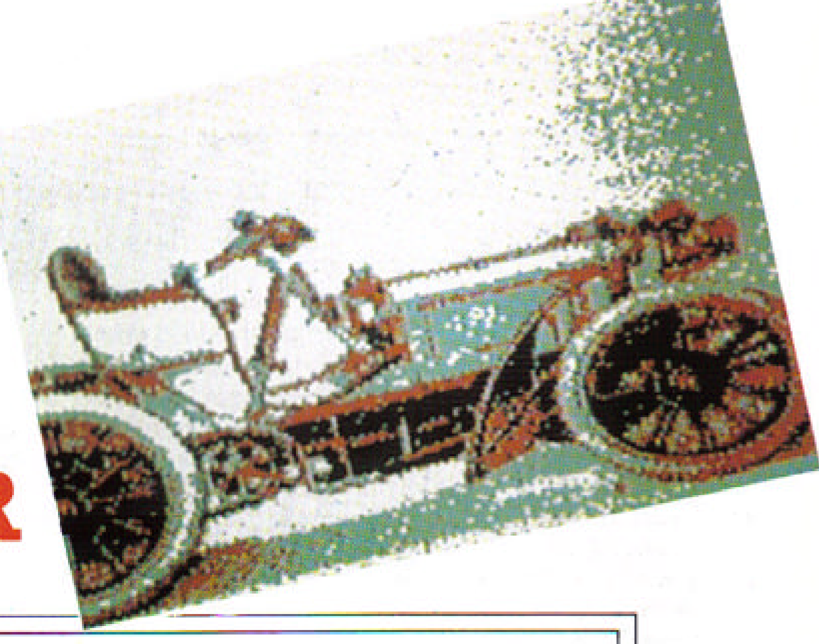
Deseo adquirir ______

Gastos envío: 300 ptas. ☐ Adjunto cheque ☐ Reembolso más gastos del mismo.

NOMBRE ______

DIRECCION ______

CONCURSO DE ARTE POR ORDENADOR



El **Commodore 64** es uno de los microordenadores del momento con mejores capacidades para la producción de gráficos. Sin duda seréis muchos los que en algún momento habéis volcado los frutos de la inspiración sobre la pantalla, utilizando fórmulas matemáticas, dibujo *pixel* a *pixel* o alguna de las diferentes herramientas de dibujo existentes.

Ahora es el momento de enviar vuestras pantallas a **INPUT Commodore,** para ver premiada vuestra creatividad.

La única limitación impuesta para la recepción de los diseños es que deben llegarnos sobre alguno de los habituales soportes informáticos, volcado de impresora, cinta, *diskette,* etc. Pueden venir acompañados de una fotografía, pero esta no es necesaria ni suficiente.

La fecha límite de recepción de las pantallas es el 30 de marzo. La decisión de nuestro jurado será inapelable.

En el número de mayo se publicarán las pantallas premiadas, quedando los derechos de copia propiedad de **INPUT Commodore.**

La presentación de trabajos implica aceptar las anteriores condiciones.



PREMIOS: microbyte

2 PRIMEROS PREMIOS:

1 tableta Koala Pad y los 11 programas de nuestro catálogo Commodore:

—Fighter Pilot
—Ant Attact
—Cave Fighter
—Slap Shot
—Anticipital
—Zaga Mision
—Cliff Hanger
—Pole Position
—Gryphon
—Black Thunder

y el programa del año en Inglaterra, Mini office (4 programas en uno): Base de Datos; Hoja de cálculo; Procesador de textos y gráficos.

2 SEGUNDOS PREMIOS:

1 Tableta Koala Pad
4 Programas Commodore:

—Zaga Mission
—Cava Fighter
—Cliff Hanger

y el fabuloso mini Office

2 TERCEROS PREMIOS

Los 11 programas de nuestro catálogo Commodore.



```
80 LINE C(K,F,0),C(K,F,1),C(
   K+1,F,0),C(K+1,F,1),1
90 LINE C(K+1,F,0),C(K+1,F,
   1),C(K,G,0),C(K,G,1),2
95 LINE C(K,G,0),C(K,G,1),C(
   K,F,0),C(K,F,1),3 .
100 NEXT K:F=1-F:G=1-F:NEXTJ
110 GOTO 110
```

Teclea para Vic 20

```
1 REM UTILIZAR CARTUCHO
 DE AMPLIACION DEL BASIC
10 GRAPHIC 1:COLOR 1,1,5,6
15 DRAW 2,0,700 TO 1023,700:
   PAINT 3,0,800
20 DIM C(100,1,1):F=0:G=1:C
   (0,0,1)=850:C(0,0,0)=28
30 S=20:L=800/S:H=L/SQR(1)
40 FOR K=1 TO S:C(K,0,0)=C(0
   ,0,0)+L*K-10+RND(1)*20
45 C(K,0,1)=C(K-1,0,1)-10+
   RND(1)*20:NEXT K
50 FOR J=1 TO S:FOR K=0 TO
   S-J
60 C(K,G,0)=20-RND(1)*40+(C(
   K,F,0)+C(K+1,F,0))/2
70 C(K,G,1)=20-RND(1)*40-H+(
   C(K,F,1)+C(K+1,F,1))/2
75 CL=(RND(1)*2)+2
80 DRAW CL,C(K,F,0),C(K,F,1)
   TO C(K+1,F,0),C(K+1,F,1)
90 DRAW CL TO C(K,G,0),C(K,G
   ,1) TO  C(K,F,0),C(K,F,1)
100 NEXT K:F=1-F:G=1-F:NEXTJ
```

El programa dibuja pequeños triángulos irregulares, partiendo desde la parte inferior izquierda de la pantalla, y construye la imagen con continuidad recorriendo toda la pantalla.

La línea 30 establece un factor de escala para los triángulos fractales, especifica la longitud de uno de los lados y su altura. La línea 40 hace un bucle de 200 pasos, estableciendo dos matrices variables para las coordenadas iniciales de cada triángilo fractal. Observa que hay un factor de aleatoriedad, de forma que los valores variarán dentro de un pequeño margen cada vez que ejecutes el programa. La línea 50 define dos bucles: uno para moverse horizontalmente por la pantalla y dibujar triángulos y el otro para desplazarse hacia arriba en la misma.

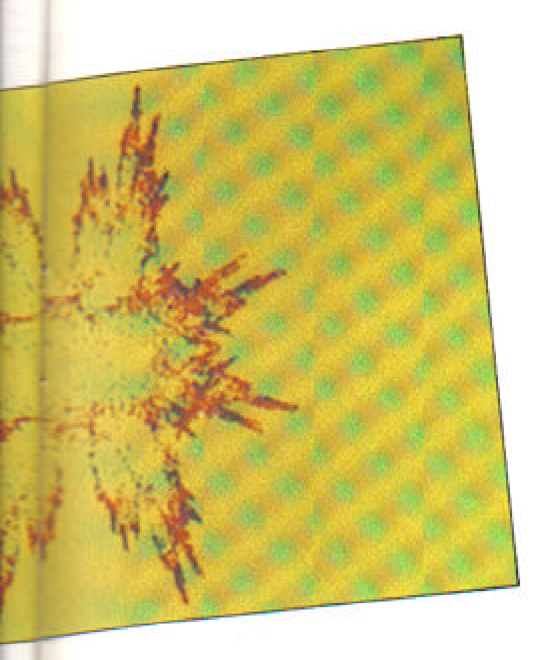
El vértice de cada triángulo se especifica en las líneas 60 (coordenada X) y 70 (coordenada Y). La línea 80 desplaza el cursor al vértice del triángulo que queda más a la izquierda y luego dibuja la base del triángulo. Observa cómo los PEEKs se restan de las coordenadas para especificar puntos absolutos sobre la pantalla; si no se hiciera esto, algunos puntos podrían quedar fuera de ella, originando un error cuando el ordenador intentara dibujarlos. La línea 90 dibuja hasta el vértice de arriba del triángulo y de aquí al vértice izquierdo, que era el punto de partida. La línea 100 completa el primer bucle para dibujar triángulos sobre la pantalla en dirección horizontal y define variables para desplazar hacia arriba la posición de dibujo y comenzar la siguiente fila de triángulos. El resto del programa sirve para llenar el primer plano, completando el dibujo.

FORMAS MULTIPLES

Ya has visto cómo cambiando los valores de algunas de las variables del programa se obtienen diferentes formas. Pero esta variación sólo afecta al tamaño, detalle y posición de la imagen y no a su aspecto global. El siguiente programa te permite especificar una gama de elementos fractales, para después construir un modelo fractal a partir de una forma en particular. El gran número de subrutinas de este listado limita el número de niveles de recursión y en consecuencia el detalle de la imagen dibujada, por lo que más adelante te presentaremos otro programa alternativo para el **Commodore 64**.

Al ejecutar el programa, la línea 230 te preguntará cuántos vértices quieres que tenga la forma que constituye el punto de partida de la fractal.

Conviene que dibujes antes la forma en un papel. Marca dos puntos que representen el principio y el final de la línea, y únelos con unos cuantos



segmentos de recta. Cuenta el número de vértices e introdúcelo como dato al programa. Sin embargo no te olvides de que tienes que especificar las coordenadas (línea 250) de todos los vértices, por lo que te conviene que sea un número pequeño; tres o cuatro son buenos valores. A las coordenadas hay que darles un valor comprendido entre 0 y 1. El bucle situado entre las líneas 240 y 280 te permite introducir coordenadas y dibujar la forma inicial. También puedes elegir si quieres que la figura especificada sea abierta o cerrada (línea 290).

La siguiente sección del programa te permite especificar la forma que ha de sustituir a cada línea recta; a esta forma se le suele llamar forma generadora. Dibújala, introduciendo la correspondiente información igual que hiciste con la forma inicial; la subrutina para introducir estos datos y dibujar la forma se encuentra entre las líneas 400 y 490. Seguidamente el programa te pedirá que especifiques el número de niveles de recursión. Después de que introduzcas el correspondiente valor, la línea 80 llama a una subrutina que determina si es la primera vez que se ejecuta el programa (en cuyo caso salta a la rutina principal, líneas 570 a 720, para dibujar las fractales), o si se trata de una nueva pasada y tiene que ofrecerte la opción de definir nuevas fractales.

PRUEBA

Como ejemplo, dale un valor de 3 como respuesta al número de vértices de la forma inicial. Seguidamente, teclea las coordenadas −0.5, −0.2 para el vértice 1; 0, 0.4 para el vértice 2; 0.5, −0.2 para el vértice 3. Si ahora respondes N cuando te aparezca el mensaje ¿CURVA CERRADA?, te aparecerá en la pantalla un triángulo sin base. Análogamente, si respondes con un 3 para el número de vértices de la forma generadora, podrías especificar las siguientes coordenadas: 0.2, 0 para el vértice 1; 0.4, 0.8 para el vértice 2; 0.6, 0 para el vértice 3. Te resultará la forma de un triángulo sin base que se asienta sobre una línea recta. Dale ahora un valor cercano a 5 como nivel de recursión y observa la fractal resultante.

Teclea para C-64

```
1 REM UTILIZAR EL CRTUCHO
  SIMON'S BASIC
10 PRINT"[SHIFT+CLR/HOME]"
20 CX=160:CY=100:IT=0
30 X=.50001:Y=0
40 GOSUB 330
50 HIRES 0,1
60 GOSUB 110
70 FOR I=1 TO 10:GOSUB 250:
   NEXT I
80 GOSUB 380
90 GOSUB 250
100 GOTO 80
110 T=Y
120 S=SQR((X*X)+(Y*Y))
130 Y=SQR((-X+S)/2)
140 X=SQR((X+S)/2)
150 IF T<0 THEN X=-X
160 RETURN
170 S=(AX*AX)+(AY*AY)
180 AX=6*(AX/S)
190 AY=-6*(AY/S)
200 RETURN
210 TX=X:TY=Y
220 X=(TX*AX)-(TY*AY)
230 Y=(TX*AY)-(TY*AX)
240 RETURN
250 GOSUB 210
260 X=1-X
270 GOSUB 110
280 IF RND(1)<.5 THEN X=-X
290 IF RND(1)<.5 THEN Y=-Y
300 X=1-X
310 X=X/2:Y=Y/2
320 RETURN
330 PRINT"ESCRIBE LOS
    FACTORES X,Y ":INPUT AX,
    AY
340 PRINT"ESCRIBE LOS
    FACTORES DE ESCALA ":
    INPUT OP
350 GOSUB 170
360 SC=2*CX/OP
370 RETURN
380 PLOT SC*(X-.5)+CX,CY-SC*
    Y,1
390 RETURN
```

Teclea para Vic-20

```
2 REM SI NO TINES EL SIMON'S
  BASIC, ESCRIBE ESTAS
3 REM LINEAS EN EL PROGRAMA
  ANTERIOR
50 POKE 53265,PEEK(53265) OR
   32
52 POKE 53272,PEEK(53272) OR
   8
54 FOR D=8192 TO 16192:POKE
   D,0:NEXT
56 FOR E=1024 TO 2023:POKE
   E,28:NEXT
380 X1=SC*(X-.5)+CX:Y1=CY-
    SC*Y
390 F=INT (Y1/8)
400 C=INT (X1/8)
410 L=Y1 AND 7
420 B=7-(X AND 7)
430 PL=8192+C*8+F*320+L
450 POKE PL,PEEK(PL) OR 2^B
500 RETURN
```

Las técnicas fractales avanzadas son las responsables de algunas de las imágenes más satisfactorias generadas actualmente por ordenador. El gran paisaje montañoso que figura en este artículo sólo existe en la memoria de un ordenador, aunque muy bien podría tomarse como algo real.

Una imagen de esta complejidad implica enormes cantidades de procesamiento por parte de grandes ordenadores especializados, pero se aplican las mismas técnicas a los ordenadores domésticos.

LA GESTION COMERCIAL RESUELTA

Cada vez es mayor el número de tareas en las que es posible utilizar ordenadores personales como herramientas. Aplicaciones que hasta hace poco eran prohibitivas por su precio para los pequeños negocios y comercios están comenzando a ser utilizadas masivamente por estos, pues cada vez es más patente que con un equipo de bajo precio (como el C64 **o el** C128**) y un programa adecuado, se puede ahorrar tiempo y dinero en tales actividades.**

Gestión Comercial es uno de estos programas. Destinado a facilitar los procesos de gestión de almacén, pedidos y facturación, está dirigido a personas con pequeños negocios que quieran comenzar a mecanizar los mismos.

El equipo necesario para poder usar el programa ha de constar, como mínimo, de un **C64** (o **C128**), una unidad de discos (**1541**, **1570** o **1571**) y una impresora (**Commodore**, compati-



ble o paralelo **Centronics**). El programa se suministra en un *diskette*, con manual de uso y un «*dongle*» (éste es un peqeño cartucho que se ha de conectar en el *port* 2 del *joystick* para que el programa funcione correctamente). El usuario puede tener tantos discos de trabajo como considere necesarios.

Además de las características antes mencionadas de gestión de pedidos, facturación y almacén, el programa ofrece una serie de posibilidades adicionales, como son la generación de informes programables, copia y creación de discos de trabajo, programación de la impresora y paso automático a contabilidad. A continuación damos una breve descripción de cada una de las funciones.

EL MENU GENERAL

Todo el programa funciona mediante menús. Esto hace que sea cómodo de manejar y fácil de aprender, sin que se requieran conocimientos especiales de informática para ello.

Al arrancar el programa entramos en el menú principal, en el que se muestran las distintas opciones disponibles. Estas son: gestión comercial, informes programables, generación del disco de trabajo, copia del disco de trabajo, conexión con contabilidad y fin de la aplicación.

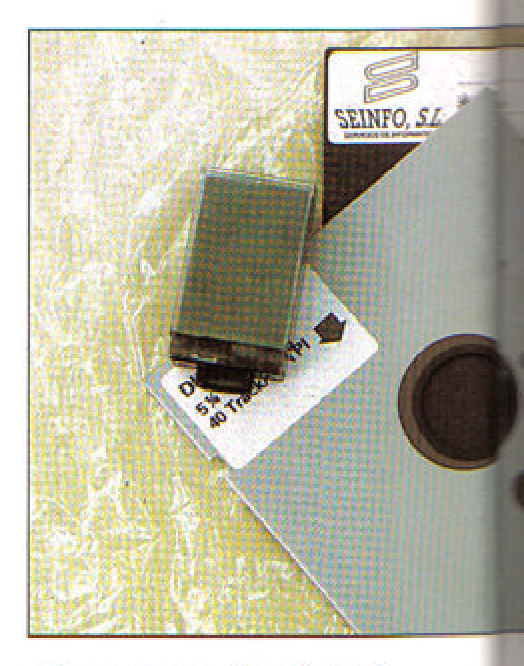
De estas cinco opciones la primera, gestión comercial, es la más importante, pues constituye el núcleo de la aplicación. Para la última, conexión con contabilidad, es necesario otro programa del mismo fabricante.

GENERACION DE DISCOS DE TRABAJO

Producir un disco de trabajo es lo primero que debemos hacer para poder comenzar a usar el programa, pues el mismo necesita un disco aparte para guardar en él los ficheros de clientes, proveedores, artículos y facturas pendientes de cierre, así como los parámetros programables por el usuario.

El disco nuevo no ha de estar formateado previamente, y el proceso de generación dura unos siete minutos.

Es en esta fase en la que el usuario



El programa en disco junto al manual y al protector.

define las capacidades máximas de los ficheros de clientes, proveedores, artículos y facturas. Las capacidades máximas son de 1000, 340, 2000 y 200 respectivamente. Sin embargo, si asignamos el máximo tamaño posible a alguno de los ficheros, ya no quedará espacio para el resto. Por esta razón, las dimensiones nunca serán las máximas indicados por el fabricante.

A título indicativo podemos decir que con 300 clientes, 600 artículos, 200 proveedores y 200 facturas se ocupa el 75% del espacio del disco.

COPIAS DEL DISCO DE TRABAJO

Permiten realizar duplicados de seguridad de nuestros discos. En realizar el proceso de copia se tardan unos cinco minutos aproximadamente.

CONEXION CON CONTABILIDAD

Existe la posibilidad de pasar directamente los apuntes que se hagan en **Gestión Comercial** a otro programa de contabilidad del fabricante. Para esto es necesario la adquisición de otro programa adicional, por lo que no analizaremos aquí este aspecto.



GESTION COMERCIAL

Como anteriormente se ha apuntado, es la parte más importante de la aplicación. Básicamente opera con tres ficheros: clientes, artículos de almacén y proveedores. Además permite almacenar cierto número de facturas pendientes de cierre.

Esta opción también presenta un menú con diversas opciones, que son las siguientes:

1) **MANTENIMIENTO DE FICHEROS**

Permite dar de alta, de baja o simplemente modificar las fichas de clientes, artículos y proveedores. También permite realizar el borrado de acumulados (ventas, compras, etc.).

Los campos que componen cada una de las fichas son:

* **Clientes**: código, nombre, dirección, población, teléfono, Banco, descuento, fórmula de pago, días de pago, CIF (código de identificación fiscal) y ventas.

* **Artículos**: código, descripción, precio de compra, precio de venta, precio medio, existencias, stock mínimo, fecha de la última compra y de la última venta y pedidos pendientes.

* **Proveedores**: código, nombre, dirección, población, teléfono, compras y observaciones.

2) **FACTURACION**

Permite elaborar facturas a los clientes, indicando descuentos, impuestos (por ejemplo, el **IVA**) recargo de equivalencia (en el caso del **IVA**), etc.

Una vez realizada la factura se actualizan las existencias del artículo facturado en el almacén. El programa nos avisa asimismo si estamos facturando por encima del *stock* disponible.

Una vez creadas todas las facturas necesarias podemos proceder al cierre de la facturación. En este proceso, además de actualizarse las ventas a cada cliente, se emiten los recibos correspondientes a las facturas, el Diario de Ventas y el Diario de Recibos. También se puede hacer el paso a contabilidad si se dispone del programa de conexión.

3) **ALMACEN**

Esta opción permite controlar fácil y cómodamente nuestras existencias de almacén, así como realizar la gestión de pedidos. También podemos llevar registro de las entradas y salidas de artículos, los primeros especificando el proveedor de los mismos.

Asimismo podemos controlar los pedidos pendientes, anotar los nuevos,etc.

4) **UTILIDADES**

Esta opción sirve para posibilitar la modificación de una serie de variables, entre ellas la máxima cantidad de fichas de clientes, artículos, proveedores y ficheros pendientes.

Adicionalmente podemos definir otra serie de parámetros que son utilizados por el resto del programa: número de la última factura, nombre del impuesto (generalmente el **IVA**), porcentajes de aplicación del mismo, recargo de equivalencia, etc.

Del mismo modo podemos modificar los párametros de la impresora (**Commodore**, compatible o paralelo **Centronics**).

INFORMES PROGRAMABLES

Esta opción del menú principal permite al usuario generar una serie de programas que se encarguen de listar las fichas de clientes, artículos o proveedores en la forma que el mismo desee. La salida se puede dirigir a la pantalla o a la impresora.

Para indicar la forma en que quiere que se listen los informes, el usuario dispone de unas cuantas reglas muy fáciles de aplicar y de una serie de parámetros que puede modificar (longitud del texto, del papel, etc.).

Una vez que se ha creado un programa de listados se puede guardar el mismo en disco, de donde podrá ser llamado tantas veces como se crea conveniente.

La programación de listados es una herramienta potente, flexible y de fácil uso, que puede permitir, por ejemplo, imprimir etiquetas con las direcciones de los clientes, listar artículos con un cierto código, productos, diferencias, etc.



CONCLUSIONES

Gestión Comercial es un programa potente, sencillo y fácil de usar que puede constituir una excelente ayuda en el control de facturación, almacén y pedidos. En nuestro caso lo hemos probado con la unidad de disco 1541 y, a pesar de la proverbial lentitud de ésta, ha resultado ser bastante veloz en todas sus opciones.

El único defecto que se le puede, quizás, achacar, es una cierta falta de claridad en el manual de uso. En nuestra opinión, un manual ha de ser una guía exhaustiva que enseñe al usuario todos y cada uno de los pasos que ha de dar al manejar el programa. El manual de **Gestión Comercial**, aún explicando todos los aspectos del programa, adolece de esta falta de detalle que nosotros consideraríamos deseable.

LOS MEJORES DE INPUT COMMODORE

PUESTO	TITULO	PORCENTAJE
1.°	Skyfox	19,6 %
2.°	Winter Games	17,2 %
3.°	Rambo	12,4 %
4.°	Hardball	11,6 %
5.°	Commando	8,8 %
6.°	Misión Imposible	7,6 %
7.°	La ley del Oeste	6,8 %
8.°	Karateka	5,6 %
9.°	Pitstop II	5,2 %
10.°	The Goonies	5,2 %
		100 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Marzo de 1986



SOFTACTUALIDAD

SILENT SERVICE



Silent Service es un programa de estratégia de **Microprose**. Recrea las aventuras de la tripulación de un submarino en las fechas de la segunda guerra mundial. Unos gráficos excelentes nos permiten ver simultáneamente varias pantallas entre las que se encuentran la sala de mando, el esquema de los daños sufridos por el submarino... etc.

****

EL CORAZON DE AFRICA



Electronic Arts lanzó en la reciente feria de Las Vegas un juego muy similar a **Siete Ciudades de Oro**. Las aventuras de un explorador en AFRICA, con escenarios que recrean el continente negro.

****

JET

Jet es la última novedad de **Sublogic**. Constituye una especie de continuación de **Flight Simulator II**. **Jet** es un simulador de vuelo de enorme potencia, que emula las evoluciones de los F-16 y F-18. Tal y como sucede en la realidad, el avión puede despegar desde un portaviones (por ejemplo **Nimitz**) o desde tierra. Es posible ver tu reactor desde diversas perspectivas, con *zoom* y hasta observar el esquema del avión... y claro está enfrentarte con los célebres **Mig**.



****

ESTACION ESPACIAL



La tristemente accidentada nave **Columbia** toma un papel principal en una novedad de la casa **Bantham**. **Space Station** muestra la vida en una estación espacial construida desde la nave **Columbia**.



Tienes que ocuparte de todo, desde la construcción en la Tierra de la estación orbital hasta el lanzamiento de la nave y puesta en órbita de la estación.

****

LA CAVERNA DEL TIEMPO



Cave of Time pertenece a la misma casa que **Escape**. Un *gnomo* te ayuda a viajar por el tiempo y has de lograr salir de situaciones en épocas



conflictivas, como la de estar rodeado por prehistóricos *mamouts* o el asesinato del presidente **Kennedy**.
Lo único que puede suceder es que trastornes toda la historia, evitando determinados acontecimientos o adelantándolos (por ejempló proporcionando música *rock* a los hombres prehistóricos).

★★★★★★★★★★★★★★★★

HUIDA (ESCAPE)

Este programa es el relato de una fuga suicida. Los Estados Unidos, tras una sangrienta guerra civil, han quedado divididos en tres territorios, de los cuales uno es una férrea dictadura.
Espías del territorio vecino han sido apresados y han de fugarse portando unos microfilmes de capital importancia.
Este juego es de una casa americana independiente y es el primero que incorpora un simulador de vuelo y otro de conducción de un *jeep*.



★★★★★★★★★★★★★★★★

HACEDOR DE JUEGOS

Activision presentó en Las Vegas un generador de juegos llamado **Games Maker**. Esta fabulosa herramienta te permitirá crear *sprites* (expandidos, multicolor, etc...), generar pantallas y sonidos e incluso dotar movimiento al resultado obtenido.
Activision confía en obtener buenos resultados con este programa y su publicidad anuncia un concurso de programas hechos con esta herramienta, cuyo premio es el lanzamiento del juego ganador.



¿Tienes un COMMODORE 64?

Para todo el que posea uno de estos magníficos ordenadores personales, la oportunidad de estar más al día ha llegado, por fin. **CLUB DE ELITE**, el mejor soft para los lectores de **INPUT**, te ofrece a partir de ahora lo mejor y más actual del software mundial, siempre con excepcionales ofertas.

PIDE LAS 3 NOVEDADES DEL MES Y PAGA SOLO 2
¡Ahórrate 2.500 ó 2.300 ptas.!
¡A TU ELECCION!

¡Todo son ventajas en este CLUB!

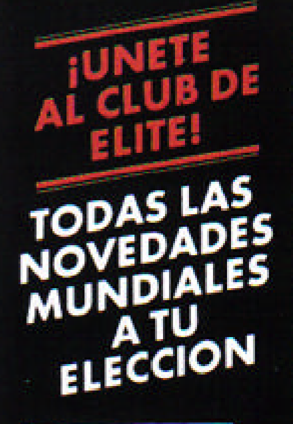
- *No hay cuota alguna de entrada. Basta ser lector habitual de INPUT COMMODORE.*
- *No hay obligación de compras mínimas por año.*
- *Podrás comprar soft (programas y/o juegos individualmente), siempre con un **10% de DESCUENTO**, o acogerte a la oferta del mes de 3 JUEGOS AL PRECIO DE 2.*

UN GRAN CLUB PARA TI, AMIGO LECTOR

Lo único que has de hacer es comprar INPUT COMMODORE cada mes y ver las ofertas que en este CLUB DE ELITE se ofrecerán. Las novedades de cada mes enriquecerán el fondo del CLUB, pues seguirán ofertándose, ya individualmente, en los meses sucesivos.

ENVIA EL CUPON ADJUNTO HOY MISMO

Te enviaremos los juegos por ti escogidos, ya sea la oferta del mes o cualquiera de ellos individualmente, **directamente a tu domicilio,** sin cargo adicional alguno.



ENVIA ESTE CUPON DEBIDAMENTE FRANQUEADO A:
EDISA, López de Hoyos, 141
28002 Madrid,
o bien llámanos por teléfono al (91) 415 9712

CUPON DE PEDIDO

SI, envíenme contra reembolso los juegos de la oferta del mes, cuyos números les indico en las casillas correspondientes, por los que me facturarán el precio de 2 juegos solamente.

GRATIS

SI, envíenme únicamente y contra reembolso el juego cuyo número indico en la casilla correspondiente, con un descuento del 10% sobre el P.V.P.

NOMBRE
APELLIDOS
DOMICILIO
NUM. PISO ESC. COD. POSTAL
POBLACION PROV
POR FAVOR, FIRMA AQUI

C 3-86

CONSTRUYE ESTE SINTETIZADOR DE VOZ

- LOS ALOFONOS
- EL CHIP SPO256
- CIRCUITO SINTETIZADOR
- MONTAJE
- SOFTWARE

«INPUT» balbuceó el Commodore 64 **como respuesta a los valores 12, 11, 9, 30, 13, 4 que seguían al DATA. Pero antes habíamos conectado al ordenador un circuito sintetizador de voz.**

Arrancarle sonidos inteligibles a mecanismos y circuitos electrónicos ha sido durante mucho tiempo uno de los sueños más perseguidos por el hombre, esto es: hacer que las máquinas imiten al hombre en su faceta más importante en la comunicación con sus semejantes. De hecho durante las épocas más recientes se han venido realizando pruebas con diversos resultados, utilizando distintas técnicas para sintetizar voz.

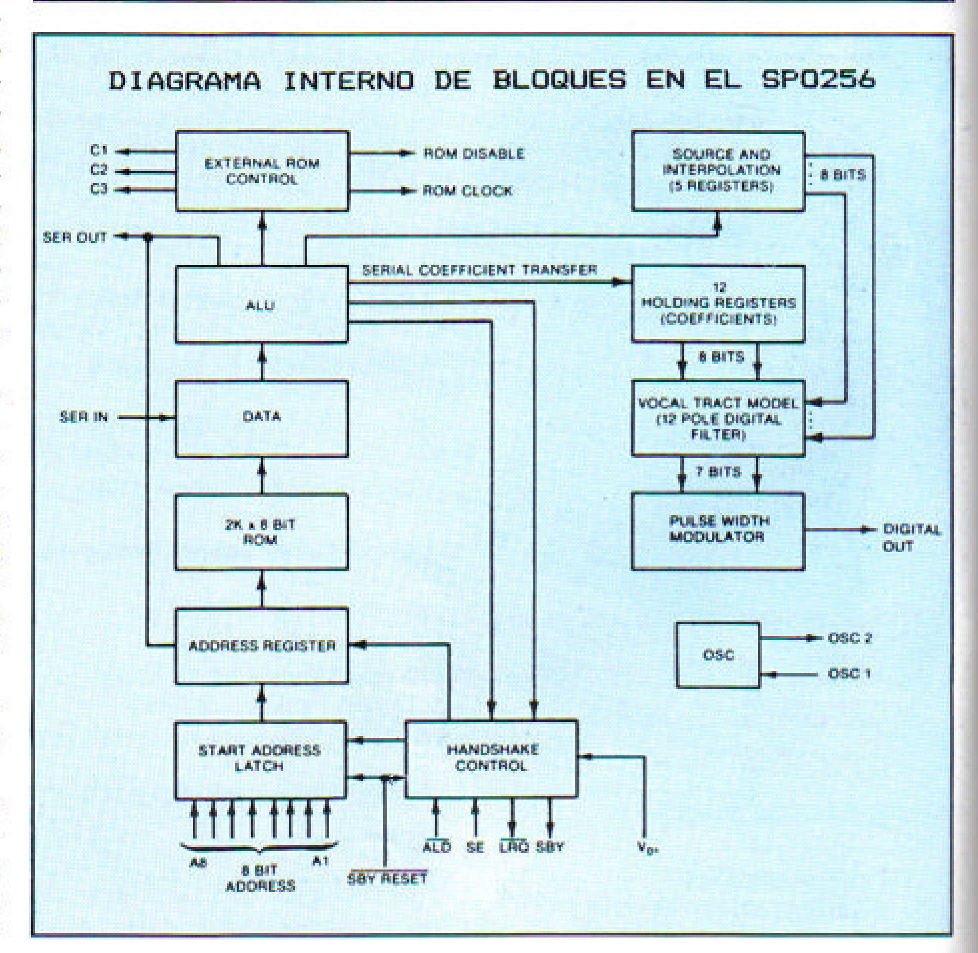
Las técnicas de producción artificial de voz pueden dividirse en dos grandes grupos: la simulación del mecanismo humano del habla y la reproducción de sonidos previamente almacenados. La primera necesita tecnología de gran complejidad para un no iniciado y la segunda precisa grandes cantidades de memoria. Un ejemplo basta para aclarar esta cuestión, para almacenar un segundo de habla medianamente legible son precisos más de cien mil bits.

Habiendo puesto nuestra vista sobre el segundo método existen dos posibilidades. La primera consiste en almacenar un conjunto limitado de palabras y posteriormente unirlas por programa para formar frases. Este es un método utilizado por varios fabricantes, entre ellos el sintetizador que lanzó **Commodore** hace más de un año sin demasiado éxito.

La segunda opción emplea alófonos, término utilizado para definir asociaciones de fonemas.

La firma **General Instruments** dispone de un *chip* de bajo costo, llamado **SPO256-AL2**, que incorpora un conjunto de 64 alófonos tomados a partir de 40 fonemas. Desgraciadame-

C-64		SPO256	
Terminal	Leyenda	Leyenda	Patilla
1	Masa	Masa	1,10,11,22
2	+5v	Vdd	7,19,23
C	PB0	A1	18
D	PB1	A2	17
E	PB2	A3	16
F	PB3	A4	15
H	PB4	A5	14
J	PB5	A6	13
K	PB6	RESET	2
L	PB7	SBY	8
M	PA2	ALD	20



SOLO EN EL COMBATE

Kung Fu Master, el popularísimo juego de las máquinas de los bares está por fin al alcance de nuestro micro, conservando todas sus características.
Tienes que rescatar una princesa que se halla aprisionada en un templo de 4 plantas. Has de enfrentarte con los ocupantes del templo: *gnomos*, mujeres que lanzan cuchillos y otros practicantes del *kung fu*... El acceso de una planta a otra está protegido por un monstruo de características siempre diferentes. Asi, en la primera planta habrás de situarte justo a su lado y golpearle con los pies, el monstruo de la segunda fase es un peligrosísimo jorobado que lanza *boomerangs*. Quizás el peor de todos sea el gigante de la tercera planta, al que habrás de atacar en los pies para poder derrotarle.
No hemos logrado vencer al mago de la cuarta planta, que se multiplica en cuanto le golpeas aparece trás de ti.





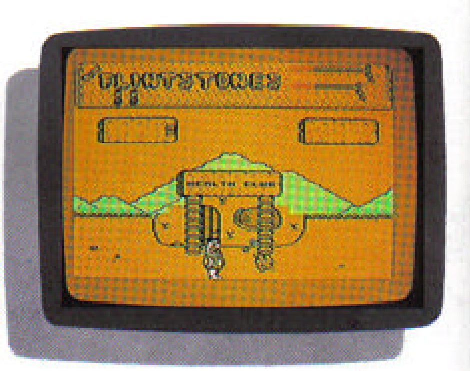
DATOS GENERALES	CALIFICACION
TITULO: Kung Fu Master	GRAFICOS: 5 sobre 5
FABRICANTE: US Gold	COLOR: 4,7 sobre 5
ORDENADOR: C-64/C-128	PRESENTACION: 4,8 sobre 5
MEMORIA: 64K	INTERES: 4,7 sobre 5
CLASE DE PROGRAMA: Artes Marciales	REALISMO: 4,8 sobre 5

****

LOS PICAPIEDRA

Estamos ante una avalancha de juegos basados en películas. Ahora es el simpático **Pedro Picapiedra** quien llena nuestros ratos de ocio.
El simpar Picapiedra tiene que convercer a **Wilma** para que forme un hogar con él. La premisa básica consiste en construirle un hogar (de piedra) para que ella se dé cuenta de quien es el hombre de sus sueños.
En un desértico paisaje **Pedro** ha de edificar la primera casa, que será el comienzo de una futura ciudad prehistórica.
Incordiado constantemente por tortugas y pterodáctilos debes ayudarle a reunir suficientes rocas, aunque mejor será localizar un troncomóvil para cargarlas.
Es importante conservar las energías de **Pedro**, porque si las pierde se queda parado.
Detalle simpático: Para colocar la chimenea, **Pedro** tiene que alquilar un dinosaurio.



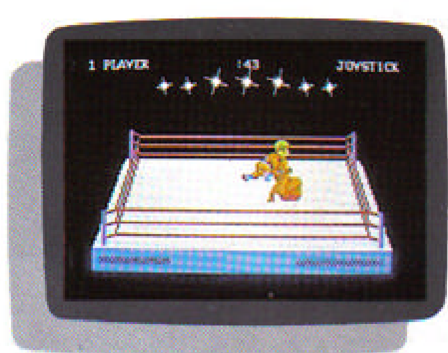


DATOS GENERALES	CALIFICACION
TITULO: Yabbadabbadoo!	GRAFICOS: 4,7 sobre 5
FABRICANTE: Quicksilva	COLOR: 4,5 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4,9 sobre 5
MEMORIA: 64K	INTERES: 4 sobre 5
CLASE DE PROGRAMA: Aventura prehistórica	REALISMO: 4,6 sobre 5

VAMOS A LUCHAR

Melbourne, la casa que se hiciera popular gracias a **Exploding Fist** vuelve a la carga con un atractivo juego de lucha.
Este programa ha tenido una enorme expectación debido a una fuerte campaña publicitaria en su pais de origen.
Dispones de un surtido de 25 movimientos básicos, procedentes de la lucha libre en la modalidad *catch*, para someter a los contrincantes que irán interponiéndose en tu camino hacia la gloria. La verdad es que antes de jugar, perdón luchar, es recomendable leer detenidamente las instrucciones. De lo contrario corres el riesgo de que te machaquen antes de lo necesario.
Desde el punto de vista gráfico el fabricante tuvo que optar entre el compromiso de mejorar los gráficos, haciendo que el programa fuese de carga secuencial, o bajar el nivel de los mismos para que el programa entrase de una vez en la memoria del ordenador.
Finalmente se eligió la segunda, a nuestra forma de ver la más acertada y el juego no se resiente demasiado. Esta pérdida es mínima y sin embargo mantiene todo el interés habitual de los juegos de lucha, siendo, sin duda alguna, el juego más completo de los existentes en lo que a cantidad de movimientos afecta.
Por si fuera poco, los nueve contrincantes están caracterizados (hay desde *punkies* hasta *pararockers*...). La velocidad y cronometración del movimiento tienen un aspecto totalmente realista. Un sintetizador de voz va contando el tiempo para decidir quien es el ganador, como en los combates reales.





DATOS GENERALES	CALIFICACION
TITULO: Rock'n'Lucha	GRAFICOS: 5 sobre 5
FABRICANTE: Melbourne	COLOR: 5 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4,9 sobre 5
MEMORIA: 64K	INTERES: 4,8 sobre 5
CLASE DE PROGRAMA: Lucha libre	REALISMO: 5 sobre 5

****

EL ZORRO DEL DESIERTO

El **Zorro del desierto** es un curiosisimo juego de estrategia y acción, que logra mantenerte en vilo durante mucho tiempo. Recrea pasajes de la segunda guerra mundial. Están observadas muchas de sus facetas: peleas con tanques, aviones, paso de tropas por desfiladeros, emboscadas, etc...
Tienes ocho opciones de juego, que comprenden desde prácticas de las distintas modalidades hasta la puesta a prueba de tus dotes estratégicas, con utilización conjunta de todas las fuerzas (aérea, naval...) en el campo de batalla. De este modo, podrás utilizar tus diferentes estaciones de radar para localizar aviones enemigos y hacer incursiones en los diferentes puntos estratégicos.
Esta es la base del juego, que reúne las ventajas de un juego de *joystick*, en cuanto a maniobrabilidad, y de

DATOS GENERALES	CALIFICACION
TITULO: Dessert Fox	GRAFICOS: 5 sobre 5
FABRICANTE: US Gold	COLOR: 4 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4,7 sobre 5
MEMORIA: 64K	INTERES: 5 sobre 5
CLASE DE PROGRAMA: Estrategia	REALISMO: 4,7 sobre 5





los juegos de estrategia, en cuanto a posibilidad de orientar el programa hacia temas de estrategia militar... Los *sprites* de los aviones recuerdan vagamente los utilizados en **Dam Busters**, pero aparte de esta característica el juego tiene un cierto aire desconocido que aumenta su interés por jugarlo.
No es fácil saber que nueva jugada tiene maquinada el mariscal **Rommel**.

****

DRAGONSKULL

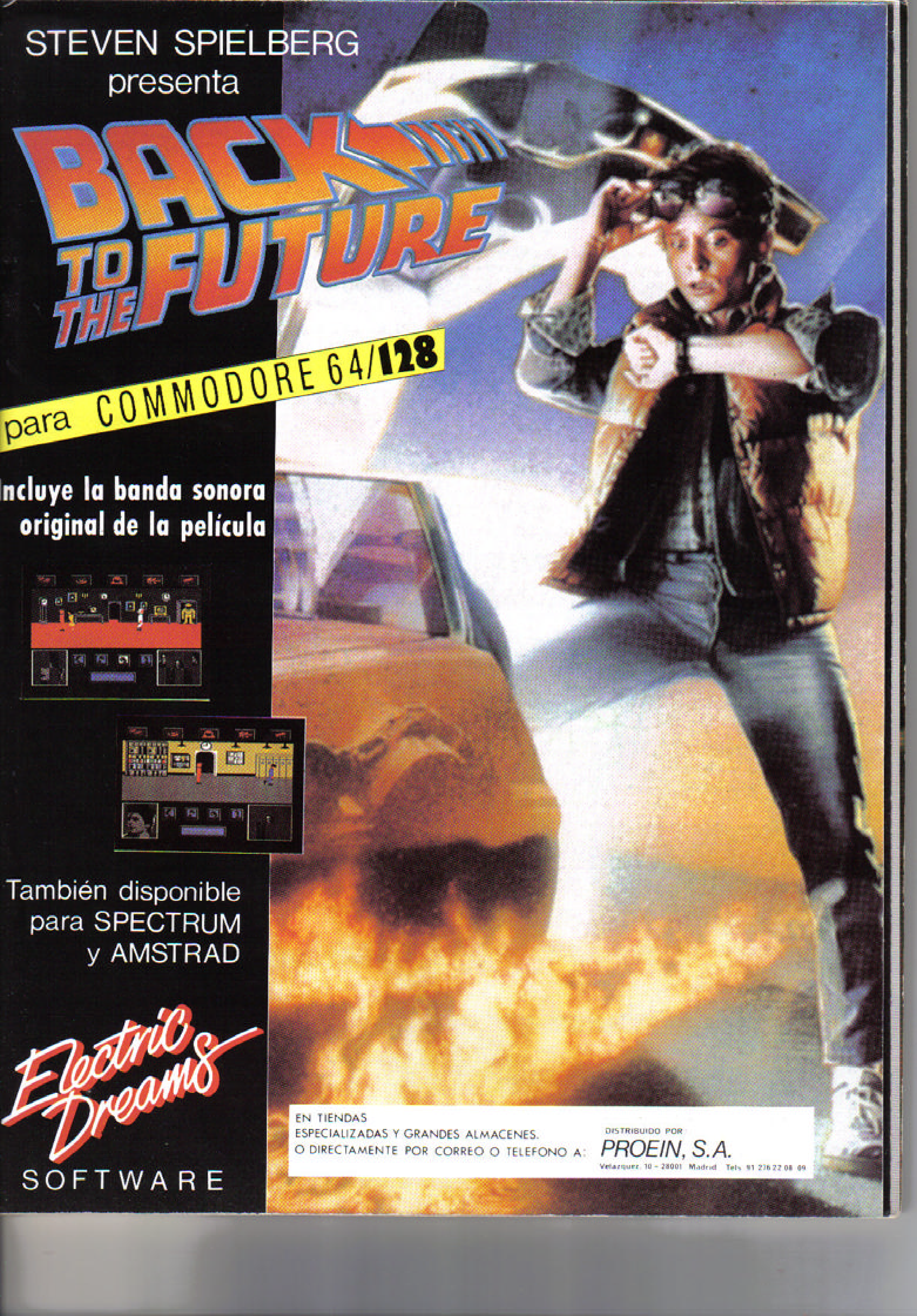
Este es el último ejemplar del aprovechamiento de una idea. Se puede comparar a las películas que continuan en segunda y tercera partes. Esto mismo puede ser aplicable a **Dragonskull**, *sir* **Arthur**, **Pendragon**, los *sprites*, el *scroll*, la definición gráfica, etc... Cuando nos enfrentamos por primera vez a este juego se deja sentir la curiosisima sensación de estar descubriendo





DATOS GENERALES	CALIFICACION
TITULO: Dragonskulle	GRAFICOS: 4,5 sobre 5
FABRICANTE: Ultimate	COLOR: 4,5 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4 sobre 5
MEMORIA: 64K	INTERES: 3,7 sobre 5
CLASE DE PROGRAMA: Aventura	REALISMO: 3,5 sobre 5

pantallas nuevas de un juego ya anterior. Esto nos hace tomar conciencia de cuanto ha mejorado el *software* de un año a esta parte.
Esta vez *sir* **Arthur Pendragon** tendrá que descubrir el secreto de la isla en la que se encuentra. Pasadizos, cuevas, gases mortíferos y muchos más serán los obstaculos que se le opondrán. Su energia menguará al contacto con unos seres diabólicos muy poco hospitalarios, que saltan sobre él a la primera de cambio.
Verás un menú de opciones aparecer en la parte superior de la pantalla.
En él podrás elegir las diferentes opciones con sólo pulsar la barra espaciadora.
Indudablemente se trata de un juego que apreciarán los incondicionales del poco afortunado *sir* **Arthur**.



RELACION DE PREMIADOS

POR PARTICIPAR EN INPUT COMMODORE

Desde la aparición de INPUT os hemos animado a enviar vuestras colaboraciones en la revista. Los trabajos publicados han sido remunerados de acuerdo con tarífas establecidas.
En este sorteo habeis entrado absolutamente todos quienes habeis "participado", aún cuando los trabajos no hayan sido publicados.
A continuación aparece la lista de los premiados en los participantes en los tres primeros meses. El mes que viene conocereis a los agraciados en los tres siguientes

NUMERO 1

Primer premio. 50.000 pts.: **Matías Salóm Avella** PALMA DE MALLORCA.
Segundo premio. 50.000 pts.: **David Sanchez Archidona** ALICANTE.
Tercer premio. 10.000 pts.: **Koldo Unanue Carazo** Beasain. GUPUZCOA.

NUMERO 2

Primer premio. 50.000 pts.: **Alejandro Manzanares Cano** BARCELONA.
Segundo premio 25.000 pts.: **Ernesto Rossa Rubio** BARCELONA.
Tercer premio. 10.000 pts.: **Alex Tarradellas Rafart** BARCELONA.

NUMERO 3

Primer premio. 50.000 pts.: **J. Francisco Aldana** MALAGA.
Segundo premio 25.000 pts.: **German Rodrigo García** Cheste. VALENCIA.
Tercer premio. 10.000 pts.: **Rafael Muñoz Alonso** BARCELONA.

Enhorabuena.

Los agraciados recibireis un vale que os permitirá comprar material informático por el valor equivalente de vuestro premio en las tiendas:

Micro Uno. c/ Duque de Sesto, 50. 28009 MADRID. Telf.: (91) 275 96 86

o

MicroTodo. c/ Orense, 3. 28020 MADRID. Telf.: (91) 253 21 19
y
MicroTodo. Centro comercial Pozuelo. Pozuelo de Alarcon. MADRID. Telf.: (91) 715 61 70.

Estos comercios os enviaran por correo el equipo solicitado a quienes no podais ir en persona.

CONTABILIDAD PROFESIONAL — 23.000
300 cuentas, 3000 apuntes (cartucho y disco)

CONTABILIDAD PROFESIONAL — 25.000
600 cuentas, 2.000 apuntes (cartucho y disco)

I.V.A. — 7.000
listados fras. emitidas y recitidas, informe contable y diario con indicacion del total a ingresar (funcinna con cartucho de contabilidad)

CONTABILIDAD + IVA — 31.000

CONTROL DE INGRESOS Y GASTOS — 7.900
99 cuentas, 250 apuntes por mes y disco, 3000 apuntes disco/año. Ideal para llevar contabilidades en Regimen de Estimacion Objetiva Singular.

CONTABILIDAD GENERAL — 7.900
segun Plan General Contable, dos niveles de cuentas, 3 y 5 digitos. Diario, mayor, balance y explotacion. 30 conceptos, 150 cuentas y 300 apuntes por cierre. (cinta)

BASE DE DATOS (disco) — 3.500

BASE DE DATOS (cinta) — 3.000

SUPERBASE 64 — 22.500
admite hasta 15 ficheros por disco, 128 campos por ficha y hasta 1024 caracteres por campo, paginas memo, etc. (disco)

SUPERBASE 128 — 26.500
igual que la anterior, pero adaptada al Commodore 128 posibilidad de 80 columnas, mas capacidad con la nueva unidad de disco 1571

HOJA DE CALCULO — 6.000
(versiones en cinta y disco)

GESTION COMERCIAL — 25.000
1000 clientes, 800 articulos, 300 proveedores, 20 representantes. (disco)

FACTURACION — 15.000
1000 clientes, 800 articulos (disco)

CONTROL STOCKS — 15.000
1000 articulos, 300 proveedores (disco)

ESCRITOR (PROCESO DE TEXTOS) — 14.900
funciona con cartucho, permite la grabacion de los datos en cinta o disco indistintamente. dispone de Ñ y Ç. Permite cartas personalizadas, etiquetas, etc. Versiones para impresoras SP800, MPS801, RITEMAN C+, STAR, etc.

PROCESOS DE TEXTO DESDE — 4.500
(versiones en cinta o disco)

SUPERGRAFICOS 64 — 4.500
50 nuevos comandos para funciones graficas, creacion y movimiento de sprites, musica y sonido etc. (versiones en cinta o disco)

SUPERASSEMBLER 64 — 4.500
editor, monitor, etc. (disco)

COMPILADOR 64 — 4.000
(versiones en cinta o disco)

AGENDA TELEFONICA — 7.900
con interface marcador automatico del numero de telefono. (Disco)

GRABADOR DE VOZ — 29.900
Incluye cartucho disco con programas y microfono.

INTERFACE COPIADOR DE CINTAS — 4.300
alta fiabilidad, permite la copia de cassette a cassette Commodore o a cassette normal, salida de sonido por el televisor.

THE FINAL CARTRIDGE — 13.900
multiplica la velocidad del disco por 5 (LOAD/SAVE) y la del cassette por 10 (todos los comandos). Interface para impresora Centronics con los caracteres del COMMODORE y volcado de pantalla en alta y baja resolucion. Monitor codigo maquina. Comandos de ayuda a la programacion. 24 K mas disponibles para basic, etc.

CALCULO DE ESTRUCTURAS — 25.000
(disco)

MEDICIONES Y PRESUPUESTOS — 25.000
(disco)

PROGRAMAS MUSICALES CON TECLADO INCLUIDO

Y LAS ULTIMAS NOVEDADES EN VIDEOJUEGOS

ACCESORIOS Y COMPLEMENTOS:

- Unidad de disco 1541 y 1571
- Impresoras
- Monitores color y fosforo verde
- Datassettes
- Tableta grafica
- Lapiz optico
- Joysticks
- Diskettes 5 1/4" 1 cara doble dendidad
- Diskettes 5 1/4" doble cara doble densidad
- Cintas cassette C-20 y C-30
- Cintas para impresora
- Fundas tipo video para cintas cassette.
- Cables conexion para impresoras
- Cables conexion ordenador/T.V.
- Cables conexion ordenador/monitor
- Papel continuo para impresora
- Etiquetas autoadhesivas en continuo

SI DESEA RECIBIR INFORMACION PERIODICA, O SI DESEA AMPLIAR INFORMACION DE ALGUNO DE ESTOS PROGRAMAS, NO DUDE EN CONSULTAR. PREVIA RESERVA DE HORA, REALIZAMOS DEMOSTRACIONES.

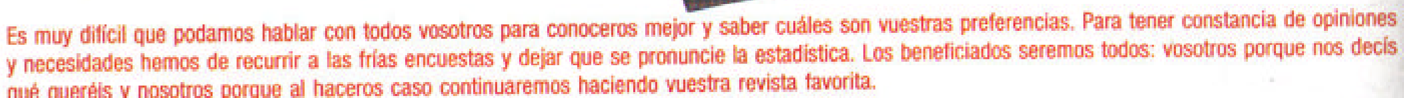
VENTAS A REEMBOLSO



Ayúdanos a ayudarte

Puedes ganar un COMMODORE 64

Encuesta para conocer tu perfil y preferencias



Es muy difícil que podamos hablar con todos vosotros para conoceros mejor y saber cuáles son vuestras preferencias. Para tener constancia de opiniones y necesidades hemos de recurrir a las frías encuestas y dejar que se pronuncie la estadística. Los beneficiados seremos todos: vosotros porque nos decís qué queréis y nosotros porque al haceros caso continuaremos haciendo vuestra revista favorita.

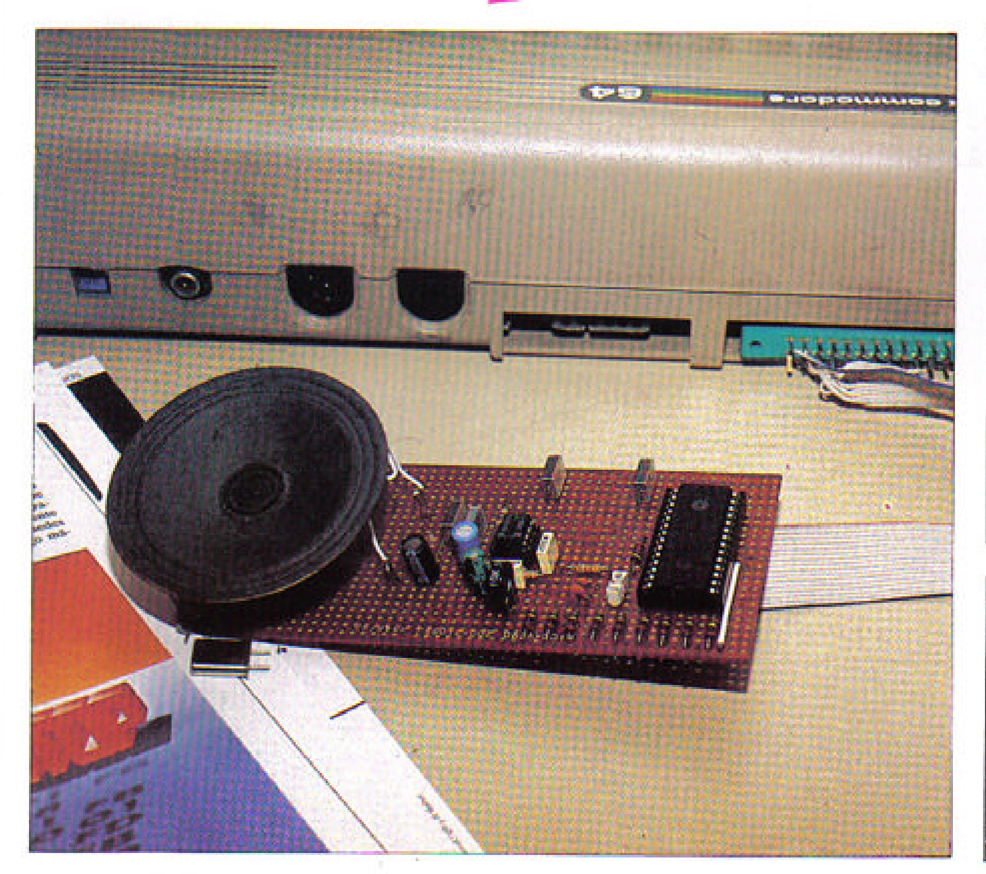
1. ¿Tienes ordenador? Sí ☐ No ☐
2. En caso afirmativo. ¿Cuál? ....
3. ¿Cuánto tiempo hace que lo posees? ....
4. ¿Qué periféricos y accesorios tienes? ....
   ....
5. ¿Desde qué número lees INPUT? N.º: ....
6. ¿Cómo conociste INPUT? ....
7. ¿La compras en kiosco? ☐ ¿Eres suscriptor? ☐
8. ¿Con qué frecuencia la adquieres?
   Todas ☐
   De vez en cuando ☐
   Otros ☐
9. ¿Cuántas personas leen tu ejemplar de INPUT? ....
10. ¿Lees toda la revista? Sí ☐ No ☐
11. ¿Tecleas?:
    Todo ☐
    Casi todo ☐
    Alguna cosa ☐
    Nada ☐
12. ¿Califica qué secciones te gustan más con mayor puntuación (10 a 0):
    Programación ....
    Código máquina ....
    Robótica ....
    Coleccionable ....
    Revista de *hardware* ....
    Revista de *software* ....
    Aplicaciones ....
    Actualidad ....
    Libros ....
    Buzón ....
    Zoco ....
    Educación ....
    Los mejores de INPUT ....
13. Califica las portadas entre 10 y 0, según lo que te hayan gustado:
    N.º 1 ....
    N.º 2 ....
    N.º 3 ....
    N.º 4 ....
    N.º 5 ....
    N.º 6 ....
14. ¿Qué artículo te gustó más del ejemplar?:
    N.º 1 ....
    N.º 2 ....
    N.º 3 ....
    N.º 4 ....
    N.º 5 ....
    N.º 6 ....
15. ¿A qué te dedicas cuando no estás con el ordenador?, ¿estudias o trabajas?, ¿en qué, si no es indiscreción?):
    ....
16. ¿Qué secciones o temas echas de menos en INPUT?:
    ....
17. ¿Estás satisfecho con tu ordenador? Sí ☐ No ☐
18. ¿Estás contento con los servicios técnico y postventa? Sí ☐ No ☐
19. ¿De qué marca y modelo será el próximo ordenador que compres? ....
20. ¿Piensas cambiarlo pronto? Sí ☐ No ☐
21. ¿Aproximadamente, cuántos programas comerciales forman tu biblioteca? ....
22. ¿Cada cuánto tiempo compras uno nuevo? ....
23. ¿Lees otras revistas de ordenadores? Sí ☐ No ☐
24. En caso afirmativo,
    ¿Cuáles? ....
    ....
25. ¿Por alguna razón en particular? ....
    ....
26. ¿Quiéres añadir alguna cosa más? ....
    ....
    ....

Con solo contestar a esta encuesta puedes ganar un COMMODORE o su valor equivalente en accesorios y/o *software*..
Así, que escribe tus datos para participar. La fecha límite es el próximo 5 de abril. No es preciso que contestes las preguntas para las que no tengas respuesta. Tampoco es necesario que las respuestas sean favorables a nuestra labor para que puedas participar en el sorteo. Critícanos si honestamente crees que debes hacerlo. Envíanos este cuestionario o su fotocopia a: ENCUESTA INPUT COMMODORE, c/ Alberto Alcocer, 46-4.º 28016 MADRID.

Nombre .... Apellidos ....
Edad .... Dirección .... Dto. Postal ....
Teléfono de contacto .... Gracias y suerte con el premio.



Accolade
LA LEY DEL OESTE
Si siempre quisistes llevar una correa con seis balas un alfiler con una estrella prendida en tu pecho, cerillas para realizar hazañas como Bat Masterson o Wyatt Earp, entonces la Ley del Oeste es tu elección. Olvida los disparos calle abajo en la calle principal. Es el año 1870 y tu eres el sheriff de Gold Guch. Tu no podrás sobrevivir a una ráfaga en el camino si en cada situación comprometida alguno de los pistoleros por un momento es demasiado rápido. Por eso usa las calles astutamente y trata de averiguar el "mal revolver" "mal disparo" de estos personajes. También debes de conservar los locales alegres. El salón el chico e incluso a tu propio sustituto. Todas son reglas que debes respetar si quieres estar vivo a media noche.
Fabricado y distribuido bajo licencia por:
COMPULOGICAL S.A.
Santa Cruz de Marcenado, 31 · 28015 Madrid · Telef. 241 10 63
LOON
SHERIFF
ANDRES SANCHEZ
Distriduido en Cataluña y Baleares por:
DISCLUB, S.A. - Balmes, 58 - BARCELONA
Tel: (93) 302 39 08 - P.V.P. 2.300 Ptas.

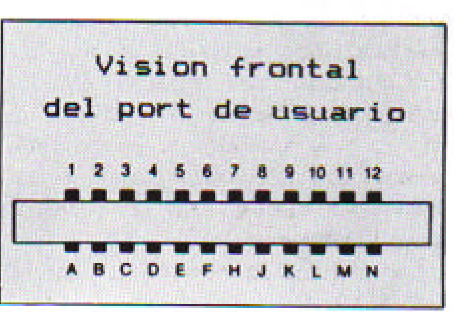


CONFIGURACION DE PATILLAS 28 patillas en DIL

Top View

Pin	Señal	Pin	Señal
1	V_{SS}	28	OSC 2
2	RESET	27	OSC 1
3	ROM DISABLE	26	ROM CLOCK
4	C1	25	SBY RESET
5	C2	24	DIGITAL OUT
6	C3	23	V_{DI}
7	V_{DD}	22	TEST
8	SBY	21	SER IN
9	LRQ	20	ALD
10	A8	19	SE
11	A7	18	A1
12	SER OUT	17	A2
13	A6	16	A3
14	A5	15	A4

Vision frontal del port de usuario

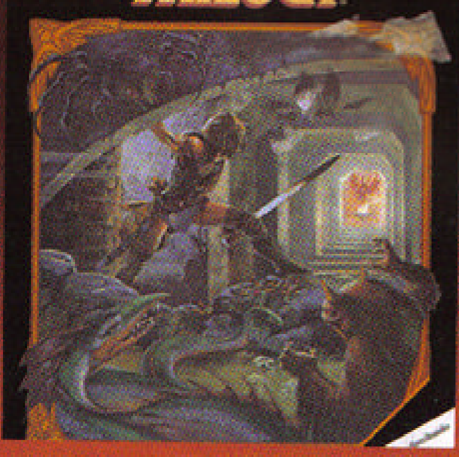


SOLO EPYX P

COMPUTER SOFTWARE



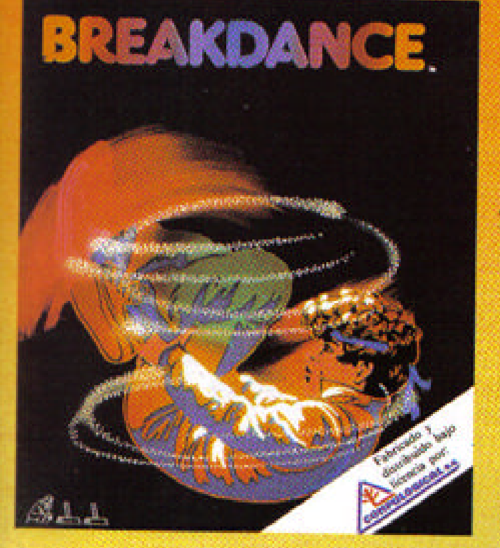
Vivimos en el año 2084. El SILICON VALLEY está envuelto en un conflicto de gran magnitud. Usted es un guerrero SILICON y su misión es programar cinco chips antes que su oponente.
• 1 ó 2 jugadores • Joystick
• COMMODORE 64.



Está ante las puertas del TEMPLO de APSHAI. Muchos aventureros han estado aquí y muy pocos han conseguido desvelar el misterio de TEMPLO, de APSHAI. Necesitará más que un carácter fuerte, inteligencia y una afilada intuición para superar la perversidad del TEMPLO.
COMMODORE 64
• AMSTRAD*
Un Jugador • Joystick.



• Soberbios Gráficos.
• Voz Sintetizada.
• Concepto original del juego.
• Juego del año 1985.
Son algunas de sus características.
Como miembro del escuadrón antiterrorista computerizado, tu misión es encontrar y alcanzar al infame ELVIN. Debes encontrar un paso a través de las habitaciones y túneles de su Cuartel General, tratando de evitar los robots protectores.
Tu misión es salvar el mundo, pero date prisa.
COMMODORE • SPECTRUM
• AMSTRAD
• 1 Jugador • Joystick.



La locura del momento, el BREAKDANCE, no debes perdértela.
Ahora cualquiera puede bailar BREAKDANCE, coge el joystick y controla tu bailarín en moon walking, stretching, popping y breaking.
Elige entre cuatro modos de juego. Un juego de acción en que tu bailarín lucha con las bandas de BREAKERS.
Intenta seguir el ritmo de tu oponente. Crea tu propia coreografía y el ordenador te hará un pleyback de ella.
Estudia la secuencia de pasos para realizar los distintos movimientos BREAK.
Aprende a bailar BREAKDANCE hoy. EPYX te lo hace fácil.
COMMODORE 64
• 1 ó 2 Jugadores • Joystick.

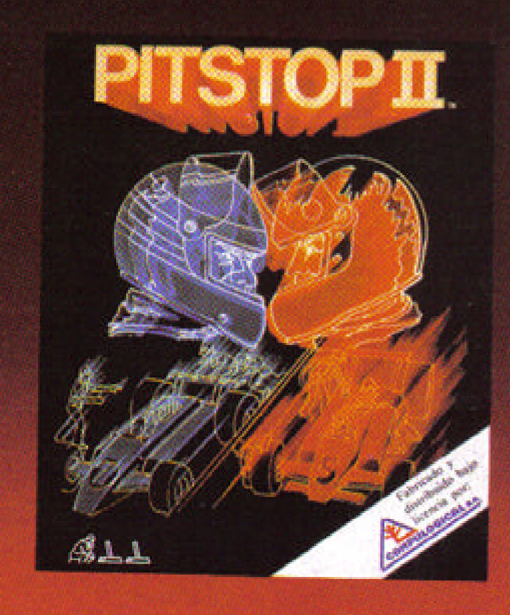
* Distribuido en Cataluña y Baleares por:
DISCLUB, S.A. - C/ Balmes, 58
08007 BARCELONA - Tel. 302 39 08

PUEDE HACERLO



Usted es un atleta olimpico compitiendo en ocho competiciones clave de los JUEGOS OLIMPICOS. Pista, tiro al blanco, gimnasia son algunas de las pruebas.

- Hasta ocho jugadores
- Control joystick
- COMMODORE 64







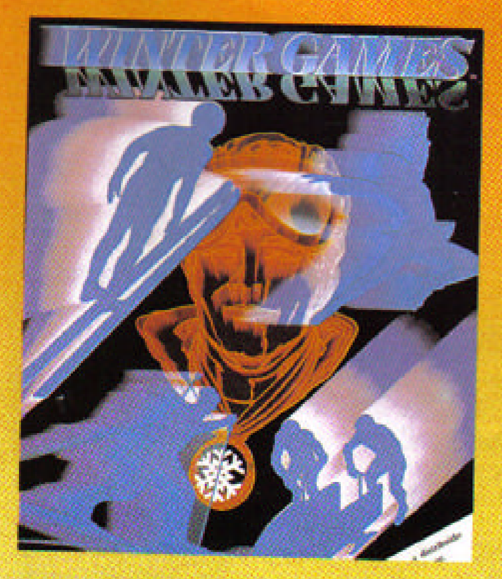
No es tarde para prepararse cara a 1988, con la dieta correcta, entrenamiento adecuado y horas de practica lo puedes conseguir.
Mientras tanto ponte el chandal, coge el joystick y deja que SUMMER GAMES II te de ocho nuevas formas de obtener el ORO.
Ciclismo, Vallas, Kayak, Triple salto, Salto de altura, Jabalina, Hipica, Esgrima son las pruebas.
Entrega de medallas al final de cada prueba. Ceremonia de apertura y cierre, son algunas de sus caracteristicas.

- Hasta 8 Jugadores
- Joystick
- COMMODORE 64
- SPECTRUM* • AMSTRAD*

- 6 circuitos diferentes.
- Excelentes gráficos.
- Escucha el rugir de los motores.
- Doble acción simultánea.

Con PITSTOP II puedes experimentar la emoción de la velocidad y el riesgo a medida que luchas con tu oponente en una carrera contra el tiempo.
Es sin duda el mejor juego de carreras existente.
COMMODORE 64

- 1 ó 2 Jugadores
- Joystick



Ha conseguido el oro en SUMMER GAMES y SUMMER GAMES II? ¡Ahora estamos en WINTER GAMES! y que increible marco para participar en sus nuevas y escitantes competiciones.

- COMMODORE 64/128
- SPECTRUM
- AMSTRAD

Fabricado y distribuido bajo licencia en exclusiva por:



Santa Cruz de Marcenado, 31 - 28015 Madrid - Telef. 241 10 63